



**NASA ECHO Project**

---

# **ECHO 10.0 Data Partner's Data PartnerGuide**

**March 2008**

**Document Version 10.3**

**Global Science and Technology, Inc**  
7855 Walker Drive, Suite 200  
Greenbelt, MD 20770 USA  
Phone (301) 474-9696  
Fax (301) 474-5970

**This page is intentionally left blank.**

## REVISION HISTORY

Date	Version	Brief Description	Author
11/1/07	10.1	Initial version for ECHO 10	Judith Wright
12/3/07	10.2	Added specific information on how to associate browse metadata to a collection/granule already loaded into the database.	Lorena Marsans
1/16/08	10.3	<ul style="list-style-type: none"><li>• Incorporated ECHO Operations' comments</li><li>• Removed "Authenticators" as they are no longer used</li></ul>	Judith Wright

## TABLE OF CONTENTS

<b>PREFACE .....</b>	<b>XI</b>
<b>REFERENCES .....</b>	<b>XI</b>
<b>CONVENTIONS .....</b>	<b>XI</b>
<b>CHAPTER 1.BEFORE YOU BEGIN .....</b>	<b>1</b>
1.1 Tasks That You Will Perform as a Data Partner.....	1
1.2 Skills You Will Need as a Data Partner .....	1
1.3 ECHO Concept and Design .....	2
1.3.1 ECHO as a Spatially Enabled Metadata Search and Order System.....	2
1.3.2 Security .....	3
1.3.3 Supported Platforms .....	3
1.4 ECHO Capability and Functionality .....	3
1.4.1 Benefits to Data Partners.....	4
1.5 ECHO Systems.....	4
<b>CHAPTER 2.THE BASICS.....</b>	<b>5</b>
2.1 Locating the ECHO Web Services .....	5
2.2 ECHO Globally Unique Identifiers (GUIDS).....	5
2.3 ECHO Entities.....	5
2.3.1 Users.....	6
2.3.2 Roles .....	6
2.3.3 Queries (Used Primarily by Client Partners).....	6
2.3.4 Catalog Items.....	6
2.3.5 Orders .....	6
2.3.6 Order Options.....	8
2.3.7 Groups .....	9
2.3.8 Conditions .....	9
2.3.9 Rules .....	10
2.3.10 Extended Services .....	10
2.4 Data Partner ECHO Functions .....	10
<b>CHAPTER 3.INGEST .....</b>	<b>11</b>
3.1 Ingest Overview .....	11
3.1.1 Changes from Previous Versions .....	12
3.1.2 Metadata Mapping/Ingest Process Considerations .....	12
3.1.3 Creating ECHO-Compatible XML Files .....	13

---

3.2	Delivering Data To ECHO For Ingest .....	13
3.2.1	Package File Delivery .....	13
3.2.2	Single File Delivery .....	14
3.3	Order of Processing .....	14
3.4	ECHO Collections .....	14
3.4.1	Collection Information .....	14
3.4.2	Spatial Data.....	16
3.4.3	Temporal Data .....	17
3.4.4	Sources and Sensors.....	17
3.4.5	Keywords .....	18
3.4.6	Additional Attributes .....	19
3.4.7	Restriction Flag for a Collection .....	20
3.5	ECHO Granules .....	20
3.5.1	Granule/Collection Association.....	20
3.5.2	Granule Information .....	21
3.5.3	Sources and Sensors.....	23
3.5.4	Additional Attributes .....	24
3.5.5	Measured Parameters .....	25
3.5.6	Orbital Information .....	26
3.5.7	Granule/Browse Association.....	26
3.5.8	Restriction Flag for a Granule .....	26
3.5.9	Spatial Representations, Coordinates and Projections .....	27
3.5.10	Two-Dimensional Coordinate System .....	27
3.6	Geometry Representations .....	27
3.6.1	Coordinate System .....	28
3.6.2	Data Types and Representation .....	28
3.7	Invalid Spatial Representation.....	35
3.7.1	Polygon Points in Counter-Clockwise Order.....	35
3.7.2	Twisted Polygon.....	36
3.7.3	Hole Crosses over Outer Ring .....	37
3.7.4	Polygon Crosses International Date Line .....	38
3.7.5	Overlapping Polygons .....	40
3.7.6	Inappropriate Data .....	41
3.7.7	Incorrectly Defined Spatial Coverage .....	44
3.7.8	Latitude/Longitude Range and Tolerance.....	45
3.8	Online Data Access URL and Online Resources URL .....	45
3.9	Browse Images and Browse Image Metadata.....	45

---

3.10 New Items vs Replace Items.....	47
3.10.1 Inserting Items .....	47
3.10.2 Replacing Items .....	48
3.11 Partially Updating Items .....	48
3.11.1 Collections.....	48
3.11.2 Granules.....	49
3.11.3 Browse .....	49
3.12 Deleting Items.....	49
3.13 Orbit Data .....	51
3.14 Ingest Reporting .....	52
3.14.1 Ingest Detail Report .....	52
3.15 Ingest Errors .....	53
3.15.1 Job Errors.....	53
3.15.2 XML Format Errors .....	54
3.15.3 Data Integrity Errors.....	56
3.16 New Items vs Update Items .....	56
3.17 Data Not Included in the Schema.....	56
3.18 Error Messages .....	56
3.18.1 Ingest Detail Report .....	57
3.18.2 Input File Validation Error Messages .....	57
3.18.3 Data Ingest Error Messages .....	60
3.18.4 Incorrect Date Format .....	66
<b>CHAPTER 4.VALIDATING YOUR METADATA.....</b>	<b>68</b>
4.1 View Dataset Information .....	68
<b>CHAPTER 5.CONTROLLING ACCESS TO YOUR METADATA.....</b>	<b>70</b>
5.1 Managing Data Access Rules .....	70
5.2 Data Access Rules.....	70
5.3 WIST Valid.....	71
<b>CHAPTER 6.CREATING ORDER OPTIONS.....</b>	<b>72</b>
6.1 Order Options .....	72
6.1.1 Assigning Option Definitions to Catalog Options.....	72
6.1.2 Elements .....	72
6.1.3 Calls .....	73
<b>CHAPTER 7.SETTING UP COMMUNICATION CONFIGURATIONS FOR ORDERS .....</b>	<b>74</b>
7.1 General Information for Configuration .....	74

---

7.2 What Must Be Specified.....	74
7.2.1 Provider Policies .....	74
7.3 Receiving and Fulfilling Orders .....	74
7.3.1 Order Fulfillment .....	74
7.3.2 Order Identifier .....	74
7.3.3 Line Items.....	75
7.3.4 User Information .....	76
<b>APPENDIX A. ACRONYMS USED IN ECHO .....</b>	<b>79</b>
<b>APPENDIX B. FUNCTIONAL BREAKDOWN BY USER/ROLE TYPE.....</b>	<b>81</b>
<b>APPENDIX C. PRE-ECHO 10 INGEST REFERENCE.....</b>	<b>87</b>
C.1 Metadata Ingest DTD.....	87
C.2 BMGT & ECHO 8/9 Partial Metadata Updates/Adds and Deletes .....	91
C.2.1 Delete Items.....	91
C.2.2 Values for the Qualified Tag for Updates, Deletes, and Inserts .....	92
C.2.3 Examples .....	94
C.2.4 Browse Insert, Update, Replace and Delete.....	96
<b>APPENDIX D. ECHO PATH URIS .....</b>	<b>101</b>
D.1 Format .....	101
D.2 Behavior .....	101
<b>APPENDIX E. ECHO ERROR HANDLING.....</b>	<b>102</b>
<b>APPENDIX F. INGEST SCHEMAS AND DTDS .....</b>	<b>104</b>
<b>APPENDIX G. BEST PRACTICES.....</b>	<b>105</b>
G.1 Tips for Preparing Your Metadata .....	105
G.2 Factors That Affect Ingest Rates.....	105
G.3 Common Errors .....	105
<b>APPENDIX H. INSERT XML EXAMPLES.....</b>	<b>106</b>
H.1 Collection .....	106
H.2 Granule.....	131
H.3 Browse.....	145
<b>INDEX .....</b>	<b>146</b>

## LIST OF TABLES

Table 1. Description of Data Partner-Related ECHO Web Services.....	5
Table 2. Supported Spatial Data Types.....	16
Table 3. Valid XPath Values for the Qualified Tags .....	92
Table 4. Browse Elements.....	97
Table 5. ECHO Path URI Behavior .....	101

## LIST OF FIGURES

Figure 1. ECHO System Context.....	2
Figure 2: ECHO Order Structure .....	7
Figure 3. The General Order Workflow .....	8
Figure 4. Ingest Overview.....	11
Figure 5. Orbit Swath.....	51

## LIST OF CODE LISTINGS

Code Listing 1. Manifest File Example .....	13
Code Listing 2. Expression of Temporal Information (RangeDateTime).....	17
Code Listing 3. Full Platform/Instrument/Sensor Description.....	17
Code Listing 4. Sample Keywords.....	18
Code Listing 5. Single Additional Attributes .....	19
Code Listing 6. Collection-Level Additional Attributes.....	19
Code Listing 7. Granule/Collection Association .....	21
Code Listing 8. Minimum Metadata Required To Insert A Granule.....	21
Code Listing 9. Granule Information.....	22
Code Listing 10. Sources and Sensors .....	23
Code Listing 11. Granule level additional attributes .....	24
Code Listing 12. ECHO 10 Additional Attributes .....	24
Code Listing 13. Measured Parameters .....	25
Code Listing 14. Orbital Information.....	26
Code Listing 15. Orbital Information .....	26
Code Listing 16. Restriction Flag Set for a Granule .....	27
Code Listing 17. Setting Two-Dimensional Coordinate System Coordinates .....	27
Code Listing 18. Geometry Example.....	28
Code Listing 19. Single Point Example .....	28
Code Listing 20. Multiple Points .....	29
Code Listing 21. Single Line Example.....	29
Code Listing 22. Multiple Line Example .....	29
Code Listing 23. Interpreted in Cartesian and Geodetic Systems .....	30
Code Listing 24. Adding Density .....	31
Code Listing 25. Single Polygon.....	32
Code Listing 26. Single Polygon with a Hole.....	33
Code Listing 27. Bounding Box .....	34
Code Listing 28. Polygon with Points in Counter-Clockwise Order.....	35
Code Listing 29. Twisted Polygon .....	36
Code Listing 30. Hole Crosses over the Outer Ring .....	37
Code Listing 31. Polygon Crosses International Dateline .....	39
Code Listing 32. Overlapping Polygons .....	40
Code Listing 33. Inappropriate Data.....	41
Code Listing 34. Incorrect Density.....	42
Code Listing 35. Correct Density.....	43

---

Code Listing 36. Polygon Covering More Than Half of the Earth .....	44
Code Listing 37. Sample Browse Insert .....	46
Code Listing 38. Sample Granule Insert with Browse Associations.....	46
Code Listing 39. Metadata Update Example.....	47
Code Listing 40. Collection Deletes.....	50
Code Listing 41. Granule Deletes.....	50
Code Listing 42. Browse Deletes .....	50
Code Listing 43. Add orbit data to Granule Metadata (ECHO format) .....	51
Code Listing 44. Add orbit parameters to Collection Metadata (ECHO format).....	52
Code Listing 45. Job Error Message .....	53
Code Listing 46. FULL_SCHEMA Error Message.....	54
Code Listing 47. STRUCTURAL_SCHEMA Error Message .....	54
Code Listing 48. SCHEMA_VALIDATION_ERROR Error Message .....	55
Code Listing 49. FILE_TYPE_INDETERMINABLE Error Message .....	55
Code Listing 50. Data Integrity Error Message .....	56
Code Listing 51. Input File Validation Error .....	57
Code Listing 52. Example of File_Type_Indeterminable Error.....	58
Code Listing 53. Invalid XML File Error .....	59
Code Listing 54. Missing Manifest File Error .....	60
Code Listing 55. Example of the Ingest Summary Report for a Successful Ingest.....	60
Code Listing 56. Error Returned When Input Data Is Older Than the Information Recorded in the ECHO Database.....	61
Code Listing 57. Error Returned When the Spatial Coverage Area Data for a Collection or Granule Is Invalid.....	62
Code Listing 58. Error Returned When Any Piece of Data Associated with a Collection or Granule Has Violated a Data Constraint .....	63
Code Listing 59. Error Returned When a Collection or Granule Identifier Sent for Deletion Does Not Exist in the ECHO Database .....	64
Code Listing 60. Error Returned Because Collection Referenced by Input Granule(s) Does Not Exist in the ECHO Database.....	65
Code Listing 61. Incorrect Date Format Error .....	66
Code Listing 62. View Dataset Information .....	68
Code Listing 63. Data Access Rules Examples .....	71
Code Listing 64. ECHO 8.0 Metadata Update DTD .....	87
Code Listing 65. BMGT Metadata Update DTD .....	88
Code Listing 66. Additional Attributes Element in Metadata Model .....	88
Code Listing 67: Removing All Online Data URLs for a Granule without Updating the Last Update for Granules Gr1 and Gr2 .....	88
Code Listing 68: Deleting a Specific Online Data URL .....	89
Code Listing 69: Insert a New Online Resource .....	89

---

Code Listing 70: Update Online Data .....	90
Code Listing 71. DTD Excerpt for Collection Metadata.....	92
Code Listing 72. DTD Excerpt for Granule Metadata.....	92
Code Listing 73. Partial Metadata Insert .....	94
Code Listing 74. Partial Metadata Delete .....	95
Code Listing 75. Partial Granule Update .....	96
Code Listing 76. Browse DTD .....	96
Code Listing 77. Sample Insert Browse XML File .....	98
Code Listing 78. Update Browse Example: testbrowseu.xml.....	99
Code Listing 79. Delete Browse Example: testbrowsed.xml.....	100
Code Listing 80: Catching Exceptions from ECHO .....	102

## PREFACE

The *ECHO 10.0 Data Partner's Guide* document provides a look and guide to the operational 10.0 release of the Earth Observing System (EOS) Clearinghouse (ECHO) from the perspective of the Data Partner.

Release 10.0 of ECHO focused primarily on Ingest, a key component of a Data Partner's functionality. This release saw a rewrite of ECHO Ingest to better utilize current technology, provide easier data upload capability, and a significant expansion of data validation prior to loading data into the ECHO repository.

The formatting of this document has been changed to make it easier to find and reference items and to increase the readability of this document both online and on paper. Changes include, but are not limited to:

- Numbering of sections for easier reference as well as making it easier to determine "where you are" in the document
- Elimination of colors and subtle font changes for section headers since colors and often fonts are not visible or obvious when printed in black and white. Additionally, colors, particularly blue, are difficult for some vision-impaired people to see or distinguish.
- Elimination of light blue for code listings and best practices/warnings.
- Addition of bookmarks for major sections and topics for easier online navigation through the document

## REFERENCES

The following documents and/or web sites were used to gather information for this document:

- ECHO 10.0 Requirements Document
- ECHO 10.0 Use Cases
- ECHO 10.0 Ingest User Guide and Reference Manual
- *The Backtrack Orbit Search Algorithm* from GeoSpatial.org by Ross Swick

## CONVENTIONS

All references to time are in Universal Time Coordinated (UTC).

Data Partners are also referred to as Data Providers.

Client Partners are also referred to as Client Developers.

Words in **bold** text are key words or concepts.

---

Programming examples (code listings) use a fixed width font (courier new) and have blue upper/lower lines separating them from the rest of the text.

---

---

Comments (denoted by // within examples)

---

*Best practices or warnings appear in italicized, boxed text.*

# CHAPTER 1. BEFORE YOU BEGIN

The EOS ClearingHOuse (ECHO) functions as a metadata clearinghouse of Earth science metadata for a wide variety of partners, enabling the science community to exchange spatial and temporal information. Data Partners provide the ECHO community with metadata representing their Earth science data holdings. ECHO technology in turn provides services for Client Partners and Data Partners and supports efficient discovery and access to Earth science data.

ECHO also functions as an order broker for the data, and offers services applied to that data. ECHO provides a portal on the internet where ECHO clients can search the metadata for information they wish to order.

Client applications can access data holdings via order distribution or online access. Data Partners retain complete control over what metadata are represented in ECHO including inserting new metadata, modifying existing metadata and removing old metadata, and controlling access to their metadata.

The primary reason for designing ECHO was to increase access to Earth science data and services by providing a system with a machine-to-machine interface, that is, an Application Programming Interface (API).

## 1.1 TASKS THAT YOU WILL PERFORM AS A DATA PARTNER

Usually performed in the order shown below:

- Providing metadata for ingest into the ECHO database
- Validating your ingested metadata
- Controlling access to your metadata through:
  - Access Control Rules
  - Setting visibility restrictions
- Creating Order Options
  - Creating order options definitions
  - Assigning option definitions to catalog items
- Setting up your (Data Provider) communication configurations for orders
  - Specifying URL and Port for order communications
  - Specifying communication retry attempts (in case of communication failure) and specifying time interval between retries

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

## 1.2 SKILLS YOU WILL NEED AS A DATA PARTNER

Since ECHO uses platform-independent web service definitions for its API, there are no requirements for a client programming language. All examples in this document are in snippets of Java code; however, the code samples provided could be translated to any web service capable language.

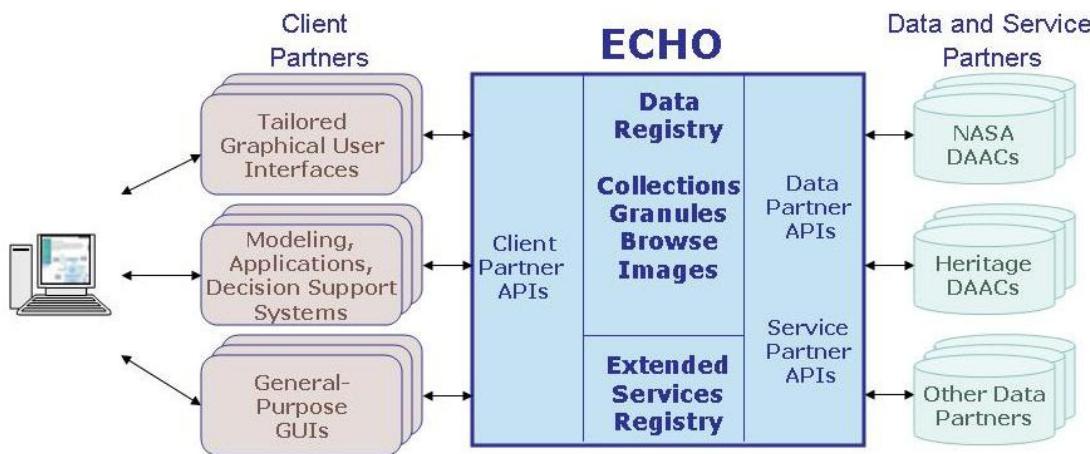
As an ECHO Data Partner, you need to be familiar with basic software development and Service Oriented Architecture (SOA) concepts such as:

- XML and XML Schema (XSD)
- Web Service Definition Language (WSDL)
- Service-based Application Programmer's Interface (API)

## 1.3 ECHO CONCEPT AND DESIGN

NASA's Earth Science Data and Information System (ESDIS) has built ECHO based on Extensible Markup Language (XML) and Web Service technologies. ECHO interfaces with different clients and users through its series of Application Program Interfaces (APIs). ECHO is an open system with published APIs available to the ECHO Development and User community.

Internally, ECHO specifies APIs and provides middleware components, including data and service search and access functions, in a layered architecture. The figure below depicts the ECHO system context in relation to its public APIs.



**Figure 1. ECHO System Context**

ECHO allows Data Partners to cache copies of their metadata within it. Data Partners have complete control over what metadata ECHO represents on their behalf. You, as a Data Partner, can insert new data, modify existing data, and remove old data.

All ECHO metadata is stored in an Oracle database with spatial extensions. The metadata model is derived primarily from that used by the Earth Observing System Data and Information System (EOSDIS) Core System (ECS). For more details about the ECHO model, refer to ECHO Earth Science Metadata Model page of the ECHO website: [http://www.echo.eos.nasa.gov/data\\_partners/data\\_model.shtml](http://www.echo.eos.nasa.gov/data_partners/data_model.shtml).

Key features of the ECHO architecture are:

- *Ease of Partner Participation* – Designed to be low-cost and minimally intrusive, ECHO offers a set of standard ways for partners to interface with the system and a metadata exchange approach that accommodates existing partners and technology.
- *Open System/Published APIs* – To accommodate independent ECHO clients, ECHO uses an open system approach and publishes domain APIs. These APIs are independent of the underlying transport protocols used. ECHO communicates using WS-I Basic Profile v1.0 compliant web services. This API is located on the Working with Web Services for ECHO on the ECHO website at: <http://www.echo.nasa.gov/reference/reference.shtml>.

Interactions with ECHO may involve user interactions in real time or may be machine to machine.

- *Evolutionary Development* – The ECHO system is being developed incrementally to allow for insight and feedback during the development cycle. Industry trends are followed and the use of commercial, off-the-shelf (COTS) products is optimized.

### 1.3.1 ECHO as a Spatially Enabled Metadata Search and Order System

Oracle enables the ECHO system to interact with spatially enabled Earth science metadata by use of spatial extensions into the system and business logic within the system that understands how to interact with that metadata. In addition, a second ECHO interface (Ingest) allows metadata updates to go directly into the

database, bypassing the message-passing API. The File Transfer Protocol (FTP) server is configured to receive these update files, which are expressed in XML conforming to three formats, one for granules (or inventory), one for collections (or datasets), and one for browse.

*Note: For ECHO 10.0 and later, these formats are schemas; for legacy ECHO, these formats are DTDs.*

See the ECHO 10.0 Ingest Formats page of the ECHO website for more information:

<http://www.echo.nasa.gov/reference/reference.shtml>

Oracle's spatial capabilities support queries for ECHO metadata whose spatial extent is described within the system. A Data Partner can define the spatial extent of a granule or a collection with different spatial constructs (for example: point and polygon). A Client Partner can then construct a search using a point, a line, or a polygon (or multiple polygon) spatial type, and ECHO responds with data whose spatial region intersects the described region.

ECHO provides services for interacting with its **Catalog** of metadata. Queries can be performed in a number of ways; result formats can be specified, and the resulting data sets can be incrementally accessed so that large return sets can be handled gracefully. ECHO also supports constructing, submitting, and tracking orders for the data that the metadata represents. ECHO supports both an embedding of a Uniform Resource Locator (URL) within the metadata for accessing the data (which the client simply accesses via Hypertext Transfer Protocol [HTTP]), and a more complicated order process in which quotes and order options are accommodated.

ECHO incorporates the ECS concept of granules and collections and defines separate schemas for updating each, under the assumption that granules will indicate which collection is considered their "primary" collection. "Primary collection" means the collection that owns the granule.

A **collection** is a grouping of granules that all come from the same source, such as a modeling group or institution. Collections have information that is common across all the granules they "own" and a template for describing additional attributes not already part of the metadata model.

A **granule** is the smallest aggregation of data that can be independently managed (described, inventoried, and retrieved). Granules have their own metadata model and support values associated with the additional attributes defined by the owning collection.

A third type of metadata, which is not spatially enabled but useful, is **browse metadata**, which provide a high-level view of granule or collection metadata and cross-referencing to other granules or collections.

### 1.3.2 Security

The ECHO system supports Secure Sockets Layer (SSL)-based communication, which a client can use to pass passwords or other sensitive information securely. Internally, the systems are firewalled to prevent unintended access.

Data Partners are encouraged to utilize SSL when communicating with ECHO during Order Fulfillment.

### 1.3.3 Supported Platforms

The ECHO system supports clients capable of initiating an HTTP connection from a variety of programming languages.

## 1.4 ECHO CAPABILITY AND FUNCTIONALITY

ECHO provides an infrastructure that allows various communities to share tools, services, and metadata. As a metadata clearinghouse, it supports many data access paradigms such as navigation and discovery. As an order broker, ECHO forwards orders for data discovered through the metadata query process to the appropriate Data Partners for order fulfillment. As a service broker, ECHO decentralizes end user functionality and supports interoperability of distributed functions.

Although this Guide focuses on the needs of Data Partners, ECHO supports the following different, nonexclusive types of Partners:

- *Data Partners* – Organizations that supply metadata representing their data holdings to the ECHO database

- *Client Partners* – Organizations that participate by developing software applications to access the Earth science metadata in the ECHO database
- *Service Partners* – Organizations that participate by advertising their Earth science-related services to the user community via ECHO, which maintains service descriptions in a Service Catalog (either special services, or services that are available as an option on a selected set of granules/collections) and support the user in ordering those services.
- *Extended Service Partners* – Organizations that participate by providing a central location for registration, classification, and maintenance of Earth science services, interfaces, GUIs, and advertisements

ECHO addresses science user needs through a set of well-defined and open interfaces upon which the user community can build its own client applications. In this way, ECHO supports extendable, flexible user interfaces, allowing industry and the science community to drive the progress of available Earth science applications. For more complete information about client applications, refer to the companion piece to this Guide, the *ECHO 10.0 Client Partner's Guide*.

The ECHO approach allows users to build their own user interfaces to ECHO, rather than being limited to the data search and order system provided by NASA. For Data Partners, ECHO offloads the burden of providing the system resources required for searching and gives users the flexibility to support community-specific services and functionality. ECHO's interoperability features allow all participants to benefit from the distributed development of functions, again reducing dependence on NASA resources.

#### 1.4.1 Benefits to Data Partners

ECHO'S open system provides Earth science data and services to a large, diverse pool of users, enabling scientific community interaction and collaboration. ECHO benefits Data Partners in the following ways.

- Control in the hands of the Data Partner
- Automate mapping between your metadata and ECHO catalog metadata
- Makes data resources available to a wide ranges of potential users
- Virtual “co-location” with other data sources and services
- Common data language
- Enable loosely coupled application solutions

### 1.5 ECHO SYSTEMS

There are three ECHO Systems that you, as a Data Partner, have access to:

- a. **ECHO Operations.** This is the current operational system for ECHO and is available to all users.  
Location: <http://api.echo.nasa.gov/echo/index.html>
- b. **ECHO Partner Test.** This is an operational system used only by the ECHO partners where they can test their data and services prior to making the final changes in the operational system  
Location: <http://api-test.echo.nasa.gov/echo/index.html>
- c. **ECHO Testbed.** This is a test system area used by partners and ECHO testers to test before changes to the ECHO system go operational  
Location: <http://testbed.echo.nasa.gov/echo/index.html>

## CHAPTER 2. THE BASICS

This chapter describes the basic terms and concepts used in subsequent discussions of the Web Services API and the ingest process.

### 2.1 LOCATING THE ECHO WEB SERVICES

To access a particular service through the ECHO Web Services API, refer to the Reference page of the ECHO website (<http://www.echo.nasa.gov/reference/>).

The table below shows each ECHO service and a brief description of its capabilities. The ECHO 9.0 Web Services API (no updates for ECHO 10.0) documentation describes in detail the services along with their operations and parameters, and is currently available online at: <http://api.echo.nasa.gov/echo/ws/v9/index.html>.

To access the Web Service Description Language (WSDL) document that describes a service, attach the suffix .wsdl from the API page following this format: <http://api.echo.nasa.gov/echo-wsdl/v9/<Service Endpoint>.wsdl>. Service Endpoints are found in Table 1.

**Table 1. Description of Data Partner-Related ECHO Web Services**

Service Name	Description	Service Endpoint
Catalog	Data warehouse searching and exploration	CatalogService
Data Management	Data Partner service to support ECHO cataloged data	DataManagementService
Group Management	Data Partner service to organize users into groups for data access control and notification	GroupManagementService
Order Processing	Data Partner-oriented service to fill and apply a status to user orders	OrderProcessingService
Provider	Data Partner account creation and maintenance	ProviderService
Taxonomy	Management interface for data and service classification schemes used by ECHO, Data Partner and Client Partners	TaxonomyService
User	User account creation and maintenance	UserService

### 2.2 ECHO GLOBALLY UNIQUE IDENTIFIERS (GUIDS)

An ECHO Globally Unique Identifier (GUID) is a mostly random number with a large number of unique keys that is assigned to an item by ECHO. A GUID is normally a 16-byte (128-bit) number in hexadecimal form.

ECHO uses GUIDs to identify items such as users, providers, contacts, orders, etc. Client applications use GUIDs to find and operate on items using the ECHO API.

In almost all cases (with the notable exception of the Taxonomy API), the GUID on an item should be null when the item is sent to ECHO to be created. Once ECHO creates the item, it will generate a new GUID for the item and return it to the client.

### 2.3 ECHO ENTITIES

This section describes several high-level concepts that help you understand the ECHO system. The following is a selected list of entities. For the complete list of ECHO entities, refer to the Ingest Input File Format.

Data Partners use the Provider User Management Program (PUMP) or the API itself to manage these entities—refer to Chapter 5, Controlling Access to Your Metadata.

### 2.3.1 Users

The most basic entity in the ECHO system is a **user**. Each user is identified by a unique user name. There are two types of users: **registered users** and **guests**. Registered users can save information they plan to use in their next session. Guests have the ability to do many of the things registered users can do, but they cannot count on persistent access to information across sessions in addition to other limitations.

To see a detailed breakdown of functionality by user type and role type within ECHO, see Appendix B, Functional Breakdown By User/Role Type.

### 2.3.2 Roles

ECHO regulates access privileges based on the concept of user roles. User roles are a way to grant a user access to the system. These roles facilitate greater flexibility with operation-level authorization and allow certain users the ability to have more than one role without having more than one account in the system.

As a Data Partner, your role is the **provider role**. You may have one or more provider roles, each of which is associated with one provider in the ECHO system. Your provider roles allow you to access and update information about the providers with which they are associated. For example, if you have a provider role for Oak Ridge National Laboratory (ORNL), then you can use the UpdateProviders operation of the ProviderService to update the contact information for ORNL.

If you have more than one provider role, you must tell the system which provider you currently want to represent. This is the "provider context". The provider context indicates the current provider that you represent, and as the name implies, it is to this provider that the provider-oriented operations in the system are applied.

*Note: If you are associated with only one provider role, the system assumes the provider associated with that one provider role is specified in the user's "provider context."*

To see a detailed breakdown of functionality by user type and role type within ECHO, see Appendix B, Functional Breakdown By User/Role Type.

### 2.3.3 Queries (Used Primarily by Client Partners)

Client Partners use queries in the ECHO system for search and retrieval of science metadata stored by ECHO. For a detailed discussion of queries, refer to Chapter 4, "Querying for Earth Science Metadata," in the *ECHO Client Partner's User's Guide*.

### 2.3.4 Catalog Items

A **catalog item** is any metadata item (granule or collection) that is available for ordering from the ECHO system. To find the desired catalog items to order, queries are performed against the ECHO database. The results of a query may return several granules or collections. Catalog items are identified by a **catalog item GUID** (CatalogItemId, which is an assigned XML metadata tag).

To see the possible required/optional options for a catalog item, invoke the **GetCatalogItemOrderInformation** operation on the **OrderManagementService** passing the catalog item GUIDs of interest.

You (Data Providers) can establish your order policy and provide ECHO with a list of your granules/collections that are orderable or searchable. Another option is for a provider to follow ECHO's rule to provide orderable notification by giving the price for each orderable item even if it is \$0.00. Note that if a URL is provided, it is assumed that the client can simply retrieve the data from that URL as a direct link or obtain the data accessing instruction via the URL. For more information about creating your order policy, refer to 5.3, Creating Order Options.

### 2.3.5 Orders

An **order** is a collection of **catalog** items that a client wishes to have and would like to order from a Data Provider. Each item in the order is associated with a quantity and any options available for that item. Within ECHO, a user

creates an order and then adds, deletes, and updates each item in the order before submitting the order to ECHO. Orders can also be created and submitted in a single web service API call (CreateAndSubmitOrder).

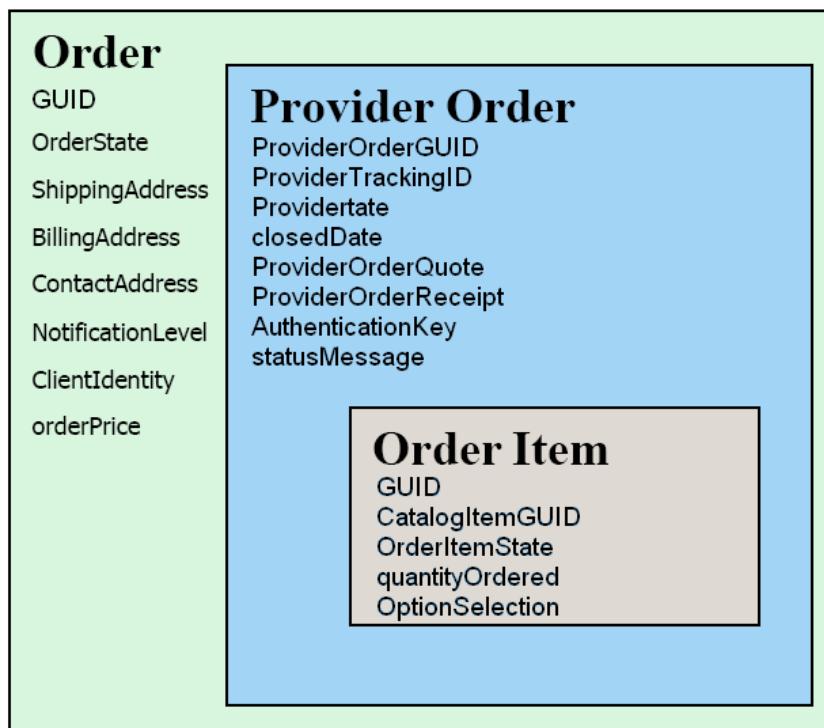


Figure 2: ECHO Order Structure

The **collection** of catalog items that comprises an order does not have to belong to one provider, but can span many providers. When organizing providers and catalog items within an order, another concept called a “provider order” is used. A provider order is a collection of all items in the original order that belong to a single provider. Since an order may contain orders from multiple providers, an order can consist of one or more provider orders. Each provider order can consist of one or more catalog items that belong to the same provider. To identify a specific provider order, you need the GUID of the order that includes that provider order and the GUID of the Data Partner associated with that provider order.

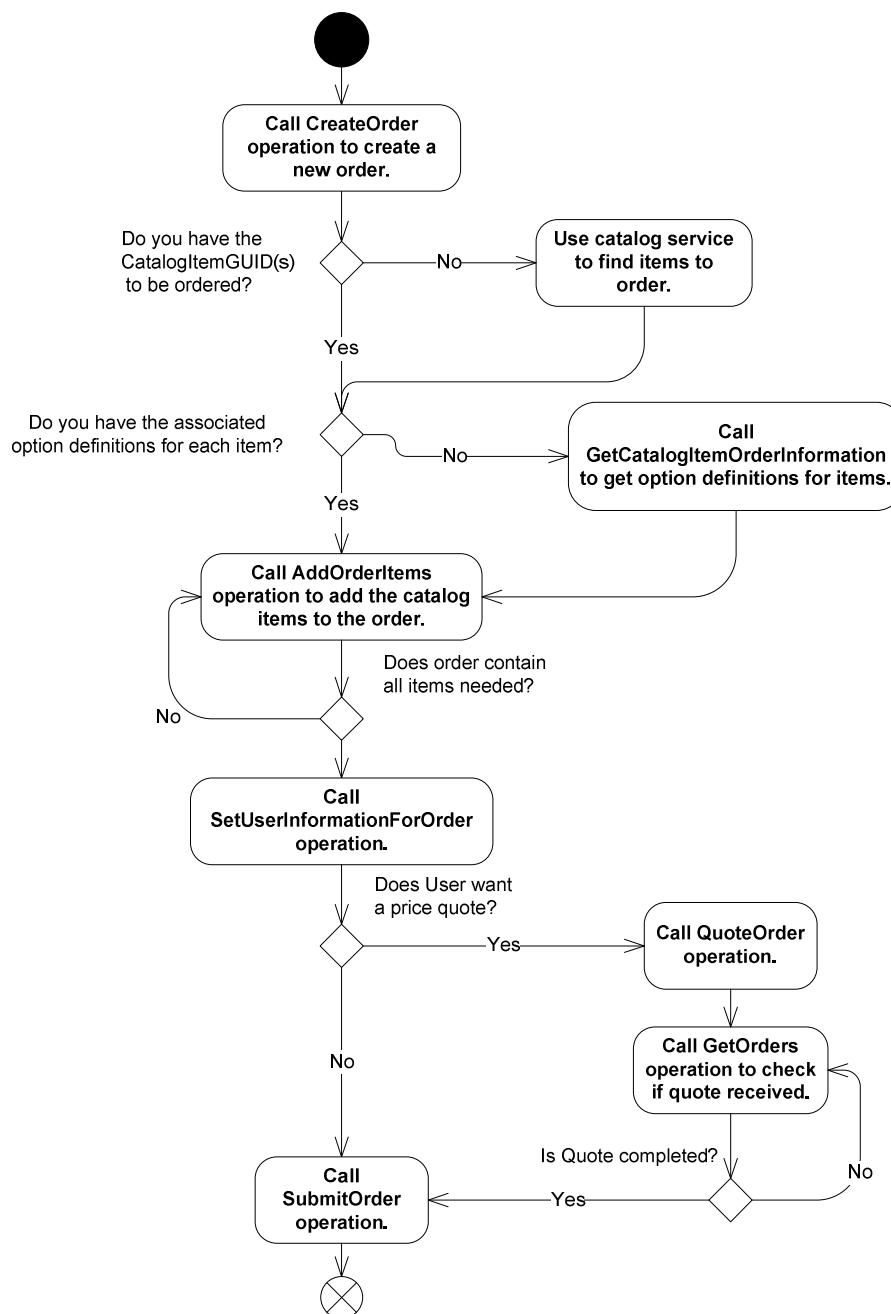
When a full order is submitted, ECHO splits the user’s order into separate provider orders and submits each provider order to the associated Data Provider.

The **OrderManagementService** allows users to create and change orders, provider orders, or individual catalog items. However, once the **SubmitOrder** operation is executed for a deRyni01order within the **OrderManagementService**, the user can no longer execute any further changes on that order. A registered user may look at the current and historical status of any of their submitted orders.

Once a provider order is submitted to the appropriate provider, the status of that order can be changed in two ways:

- The Data Provider can send an immediate response, whether they will or will not accept the order, to an order submission.
- The Data Provider can wait and asynchronously use the **OrderProcessingService** to change the status of an order after they have had time to process the order.

Most Data Providers use Provider User Management Program (PUMP) for group management, user management, provider information, provider contacts, provider policies, provider orders, data management, user lookup, and storing your address, phone numbers, and other options. For more information on the Order Management Service, refer to section 5.1, Managing Data Access Rules in Chapter 5, Controlling Access to Your Metadata.

**Figure 3. The General Order Workflow**

### 2.3.6 Order Options

Data Partners generally use Provider User Management Program (PUMP) or the API itself to set up order options. To access PUMP, go to <https://api.echo.nasa.gov/pump> for the Operational system and <https://api-test.echo.nasa.gov/pump> for the partner test system.

Use **order options** to describe the structure of the data to request of the client as well as how to display the order form to the client. **Option definitions** contain a name, scope, deprecated flag, and an ECHO Form. Every option definition for a provider has a unique name. The **scope** indicates whether it is a system-level option definition or provider-level option definition. The **deprecated flag** indicates that you, as a Data Partner, have made that

option definition obsolete. The deprecated flag indicates to clients that they should no longer use the option definition. The form part of the option definition contains an ECHO Form.

The **OptionSelection** is the data from the client for a specific option definition. The selection should be put in the content part of the option selection. Options are also used in ECHO to describe authenticators and options for ordering data from a provider.

For more information on Order Options, refer to 5.3, Creating Order Options.

### 2.3.6.1 Data Provider-specific processing options

A Data Provider may have the ability to fulfill orders in ways beyond the scope of other providers. These variations are usually associated with three broad categories.

- Distribution – physical and non-physical media types, differing formats
- Subsetting – spatial, temporal and parameter subsetting
- Higher level processing – generation of complex products from base ones

Since one size does not fit all in this arena, a generic means of presenting these additional options to the user is required.

The ECHO 10.0 Fulfillment API provides rudimentary support for generic processing options via the Order.Items.OptionSelection.Content element. This element is a simple string that can contain an XML document.

The ECHO Forms Specification Version 0.3 provides specific support for generic processing options. These options are presented as an XML document.

The contents of this XML document are intended to conform to a Data Provider-specific schema. This schema is based on an ECHO Forms Specification and is developed by the Data Provider to describe any processing options that cannot be described by the Submit API.

The ECHO Forms concept is split into two distinct concepts

- a. Defining the structure of processing options.
- b. Suggesting a user interface for user-population of that processing option structure.

For example, a Data Provider may support temporal subsetting. The structure for the processing options associated with temporal subsetting may be a start date and an end date. A Provider can not only specify that these two parameters are required for temporal subsetting, it can also suggest the type of user interface presented to the user, define rules associated with the input, and provide default values.

*See section 1.3 of the ECHO Forms Specification for more details.*

### 2.3.7 Groups

The term **groups** refers to an aggregating mechanism in ECHO that allows Data Partners to associate a Group name with a given set of users. When a group is created, the group's owner specifies an ECHO user to be the group's manager. However, group managers can be added and removed after creation by other group managers. After becoming a member of a group, a user can be granted access to restricted metadata via the Data Management Service—refer to Chapter 5, Controlling Access to Your Metadata for details.

### 2.3.8 Conditions

Conditions represent a partial equation to be evaluated as part of the Access Control (ACL) honoring system. The type of the condition defines the evaluation process. Temporal Conditions use a date range, so that a date associated with a granule can be compared against the date range to check for applicability of the condition. The primary use of conditions is to facilitate reuse among data rules. The same temporal condition can be used by both a restriction and a permission to control access to metadata. To extend a time range, you only have to change one **TemporalCondition** as opposed to changing multiple data rules. Another type of condition is **RestrictionFlag**, which can be used to restrict access to collections or granules based on the value of a

**RestrictionFlag** metadata field. For example, this might be used to control access based on a granule's science quality.

### 2.3.9 Rules

Rules include conditions and provide a complete evaluation. Rules define which specific data is to be controlled, as well as the condition to use for evaluating whether the data should be controlled. Rules also contain a comparator, which is a key part of rule evaluation. Lastly, rules contain data including **ActionType** (describes which actions the rule applies to), and in the case of a permission, a **GroupName** (describes which Group the permission applies to). **Restrictions** (one type of a rule) apply to the global ECHO population, and **permissions** (the other type of a rule) apply to a specific group. Refer to Chapter 5, Controlling Access to Your Metadata for details about using rules for data management.

### 2.3.10 Extended Services

ECHO Extended Services allows partners to advertise and register web service interfaces, implementations, GUIs, and advertisements.

## 2.4 DATA PARTNER ECHO FUNCTIONS

Functionality within ECHO is designed to provide functionality for all users including Data Partners, Client Partners, Service Partners, and the user community interested in the data, clients, and services provided by those partners. The following functionality is provided for Data Partners:

## CHAPTER 3. INGEST

### 3.1 INGEST OVERVIEW

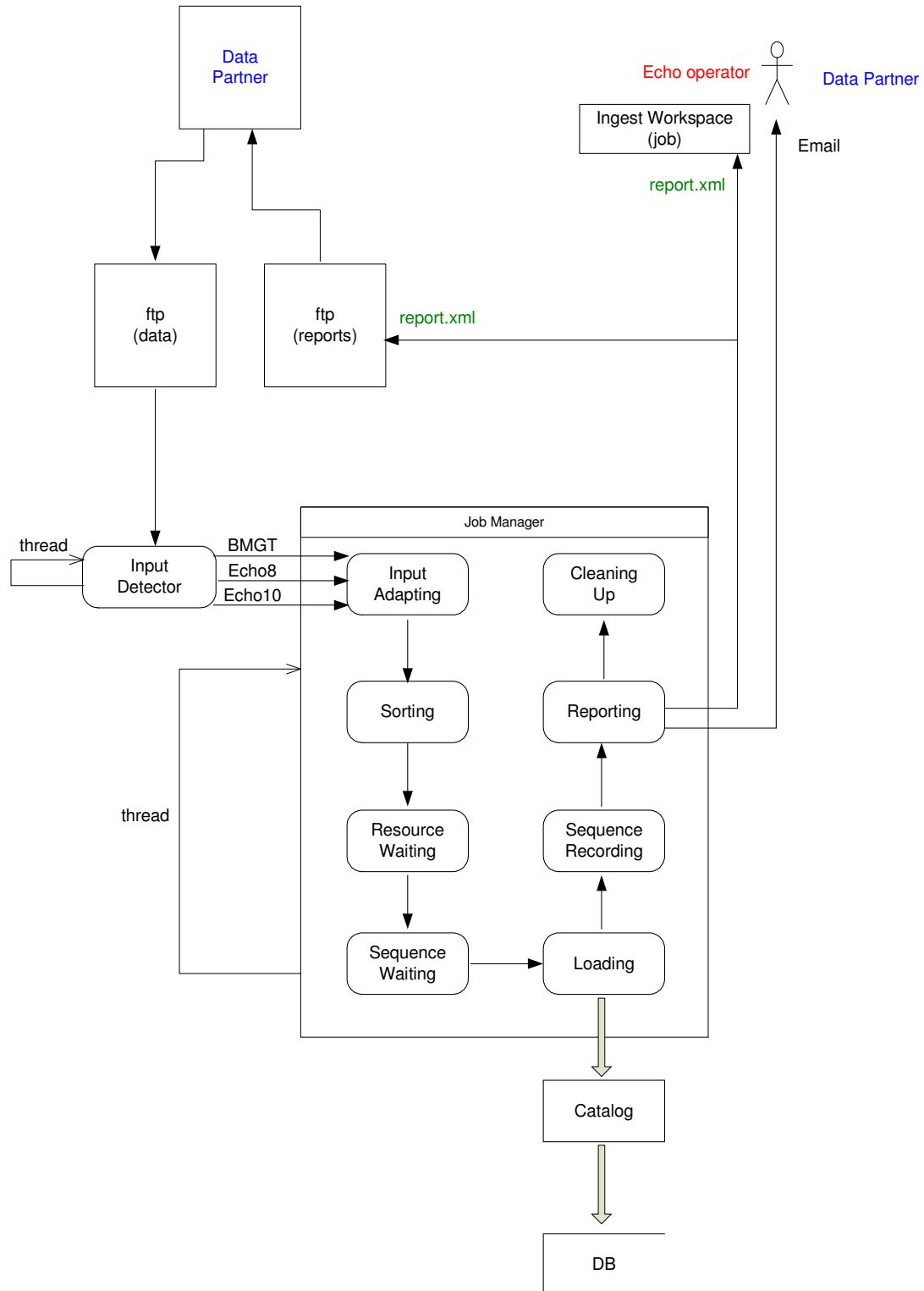


Figure 4. Ingest Overview

Metadata ingest refers to the process of inserting, deleting, or updating metadata in the ECHO database and affects only that Data Partner's specific metadata. The main tasks for the metadata ingest process are loading bulk metadata and updating the ECHO database. Although there are slight variations from one Data Partner to another in the transmission process and interaction with ECHO Operations, the process includes the following:

- a. Use FTP Push to send the metadata to a folder at `ingest.echo.nasa.gov`, the input holding area for operational Ingest. There is also a Partner Test System FTP site (`ingest-test.echo.nasa.gov`), which is sometimes used for testing metadata prior to ingest into the ECHO database.
- b. ECHO Ingest checks input folders for new or updated metadata periodically. When it detects your metadata, Ingest will automatically prepare it for updating the ECHO database. Files are aggregated into jobs and put into a job directory to await their turn to be processed by Ingest.
- c. The ingest process will add your metadata to the ECHO database immediately unless there are other jobs being processed, in which case your metadata will be placed in a "staging area" to wait its turn.
- d. When Ingest is complete, you will receive a notification, via email containing the location of the Ingest Detail Report detailing the job activities including errors, if any.

The metadata input includes Collection, Granule, Update, and Browse metadata. If you generate the input for metadata ingest using your own format, then you must convert your metadata to the ECHO Format, an XML format, before sending it to ECHO.

Before ingest can begin, ECHO Operations must assign each Data Partner a database account to host metadata. Data Partners receive their own unique provider names, determined by agreement between each Data Partner and the ECHO Operations team.

### **3.1.1 Changes from Previous Versions**

The ECHO Ingest process directly accepts and processes metadata input files conforming to the ECHO 10 Format. Unless supported by previous versions of Ingest, metadata input files conforming to other formats require that you perform a conversion before submitting the files to the ECHO ingest process. Contact the ECHO Operations team for more information.

Your FTP directory will be scanned for all types of metadata files. Unless package file delivery is used, during a single scan any number and types of metadata files may be aggregated into one job, see 3.2 Delivering Data To ECHO For Ingest for details. Within a job, metadata files will be processed in a particular order to ensure aggregated metadata files are processing correctly within a job, see 3.3 Order of Processing for details.

Finally, the ingest process will validate the input file against the ECHO 10 Format and business rules. If this validation fails, the ingest process will reject the input file if it cannot be parsed or reject only the invalid metadata items found in an input file. Whenever the ingest process rejects an input file or metadata item, it will record an error and include the error in the Ingest Detail Report. The same information will be available to the ECHO Operations staff as well.

Once metadata items in an input file have been successfully validated, the ECHO ingest process will load the metadata into the ECHO database, making the metadata available for public search. The ingest process will send an end of job email to your preconfigured provider contact email address and will send the same information to the ECHO Operations staff as well. This email will contain the location of the Ingest Detail Report for you to download.

ECHO applies additional constraints at the time of ingest to maintain metadata integrity. For a comprehensive list of the Business Rules for ingesting data, refer to the ECHO Supplementary Specification by going to the ECHO Development page of the ECHO Website and clicking on "More about Version 10.0":  
<http://www.echo.eos.nasa.gov/development/index.shtml>.

### **3.1.2 Metadata Mapping/Ingest Process Considerations**

The ECHO Ingest process directly accepts and processes metadata input files conforming to the ECHO schema. Metadata input files conforming to other DTDs require that you perform an XML-to-XML conversion before submitting the files to the ECHO ingest process. Contact the ECHO Operations team for more information.

When the ECHO ingest process detects potential input files in your FTP input directory, it will then examine and validate each file, checking to see if the first line of text contains the <!xml....> declaration.

- **If this line is not present**, the ingest process will reject the input file.
- **If this line is present**, the ingest process will verify that the file is located in the appropriate directory. For example, if you have placed a granule XML file in the collection (rather than the granule) FTP input directory, the ingest process will reject the input file.

Finally, the ingest process will validate the input file with its corresponding DTD. If this validation fails, the ingest process will reject the input file. Whenever the ingest process rejects an input file, it will record an error and send it to you via your preconfigured provider contact email address. It will send the same information to the ECHO Operations staff as well.

Once an input file has been successfully validated, the ECHO ingest process will load the metadata into the ECHO database and update the operational database tables, making the metadata available for public search. The ingest process will send you an end of job email to your preconfigured provider contact email address and will send the same information to the ECHO Operations staff as well. This email will contain the location of the Ingest Detail Report for you to download.

ECHO applies additional constraints at the time of ingest to maintain granule metadata integrity. For a comprehensive list of the Business Rules for ingesting data, refer to the ECHO Supplementary Specification by going to the ECHO Development page of the ECHO Website and clicking on "More about Version 10.0": <http://www.echo.eos.nasa.gov/development/index.shtml>.

### 3.1.3 Creating ECHO-Compatible XML Files

Generate your metadata XML files using the ECHO 10 Format. ECHO has its own schemas defined for collection, granule, and browse metadata, which you can download from the Reference page of the ECHO website by selecting "10.0 Ingest Formats" under "Current System" (<http://www.echo.nasa.gov/reference/reference.shtml>).

Along with metadata structure, the ECHO 10 format specifies certain data characteristics such as data type, length of field, etc. If metadata files received do not meet the specifications described in the ECHO 10 Format, those items will be rejected. Data validation is further performed during processing of the metadata itself.

## 3.2 DELIVERING DATA TO ECHO FOR INGEST

### 3.2.1 Package File Delivery

Package delivery was introduced with ECHO Ingest 10.0 and is the preferred delivery mechanism for providers. You (the data provider) deliver a package that is a ZIP archive of metadata files. The archive contains metadata files of any action and type as well as a manifest file that provides a list of the files in the package and an optional sequence number.

Packages have the benefit of compression, which greatly reduces the amount of data delivered. They also support optional sequencing that allows you to indicate the order that the packages should be executed in—which is important in the event that packages arrive out of order and have dependencies on other packages.

ECHO Ingest guarantees that a single package will map to a single job with all the files in the package included in the job. Browse binary files are still delivered externally to the archive due to the size and number of the files involved.

#### Code Listing 1. Manifest File Example

---

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Manifest packageName="1101" sequenceNumber="1">
  <Files>
    <File name="CollectionInsertFile.XML"/>
    <File name="GranuleInsertFile.XML"/>
```

---

```
</Files>  
</Manifest>
```

---

*Note: The manifest file for package delivery must be named "manifest.xml"...all lowercase letters.*

Refer to the package manifest schema and documentation for more information on constructing a package.

### **3.2.2 Single File Delivery**

Single file delivery is the legacy delivery mechanism that most providers currently use. This mechanism allows you to deliver uncompressed metadata files one at a time. ECHO Ingest will monitor the delivered files for completion based on arrival date. Once a file is determined to be complete, a job will be constructed and the file added.

ECHO Ingest will attempt to put as many files into a single job as possible, but cannot guarantee that all of the delivered files will be in a single job because some files may not be considered complete when the input scan is performed. It is therefore important for you to deliver the files serially in the order that they desire execution. Once a job is created, ECHO Ingest will generate an internal sequence number to ensure that individual jobs are executed in the order they were created.

## **3.3 ORDER OF PROCESSING**

ECHO Ingest will process input found in a single job in the following order:

- a. Browse inserts/replacements
- b. Collection inserts/replacements
- c. Collection partial deletes
- d. Collection partial updates
- e. Collection deletes
- f. Granule inserts/replacements
- g. Granule partial deletes
- h. Granule partial updates
- i. Granule deletes
- j. Browse deletes

## **3.4 ECHO COLLECTIONS**

The complete ECHO Collection format is available on the Reference page of the ECHO website by selecting "10.0 Ingest Formats" under "Current System" (<http://www.echo.nasa.gov/reference/reference.shtml>).

### **3.4.1 Collection Information**

- a. The required elements for collection metadata are:
  - 1) DataSetId
  - 2) Description
  - 3) InsertTime
  - 4) LastUpdate
  - 5) LongName
  - 6) Orderable

- 7) ShortName
- 8) VersionId
- 9) Visible
- b. Additional (Optional) Elements
  - 1) AdditionalAttributes
  - 2) AlgorithmPackages
  - 3) ArchiveCenter
  - 4) AssociatedBrowseImages
  - 5) AssociatedDIFs
  - 6) Campaigns
  - 7) CitationForExternalPublication
  - 8) CollectionAssociations
  - 9) CollectionState
  - 10) Contacts
  - 11) CSDTDescriptions
  - 12) DataFormat
  - 13) DeleteTime
  - 14) DisciplineTopicParameters
  - 15) MaintenanceAndUpdateFrequency
  - 16) OnlineAccessURLs
  - 17) OnlineResources
  - 18) Platforms
  - 19) Price
  - 20) ProcessingCenter
  - 21) ProcessingLevelDescription
  - 22) ProcessingLevelId
  - 23) RestrictionComment
  - 24) RestrictionFlag
  - 25) RevisionDate
  - 26) Spatial
  - 27) SpatialInfo
  - 28) SpatialKeywords
  - 29) SuggestedUsage
  - 30) Temporal
  - 31) TemporalKeywords
  - 32) VersionDescription

The metadata in these elements can help describe a provider's collection metadata for use in searches.

---

ECHO requires that you provide a **DataSetId**, which is a unique identifier, to identify a collection. In most cases, this is the **ShortName** and **VersionId**. **VersionId** refers to your (the Data Partner's) metadata ID.

**InsertTime** is the day and time you inserted the collection metadata into your database, and **LastUpdate** is the day and time you last updated your collection.

### 3.4.2 Spatial Data

Spatial metadata refers to the area of the Earth that the data cover. The ECHO system uses the Oracle database to store spatial information for collections and granules. The spatial coverage area for a granule must be within the spatial coverage area of its primary collection. To ensure that spatial searches will return the correct results, you must prepare the spatial metadata according to the detailed guidelines discussed below.

The ECHO system accepts both Cartesian, Geodetic, and Orbit coordinate systems; refer to the table on the following page for supported spatial data types and guidelines for limitations to granularity. You will choose a coordinate system based on the size and projection of the original data of the spatial area covered. You may not combine spatial types; they are mutually exclusive.

With the Oracle Cartesian coordinate system, the acceptable spatial data types include Point, Line, Bounding Box and Polygon. With the Oracle Geodetic coordinate system (World Geodetic System 84), the spatial data types supported include Point, Line, and Polygon.

**Table 2. Supported Spatial Data Types**

Spatial Data Type	Cartesian	Geodetic	Orbit	Guidelines and Restrictions
Point	✓	✓	✓	
Bounding Box	✓	✗	✗	Since neither the Geodetic nor the Orbit coordinating systems support the bounding box type, the ECHO system stores bounding box data as a polygon with four vertices (in the flat Cartesian coordinate system).
Line	✓	✓	✓	A line that has vertices across the International Date Line or across the poles is invalid spatial data for the flat Cartesian coordinate system. These restrictions do not apply to Geodetic and Orbit coordinate systems.
Polygon	✓	✓	✓	<p>A polygon's vertices must be stored in order of vertex connection. Provide the vertices in clockwise order. No consecutive vertices may have the same latitude and longitude, that is, no repeating points.</p> <p>Keep in mind that a polygon that has vertices across the International Date Line or across the poles is invalid spatial data for both the flat Cartesian and the Geodetic coordinate systems.</p> <p>No polygon should cover more than half the Earth or span half the Earth in both the flat Cartesian and the Geodetic coordinate systems.</p>

The Oracle Geodetic model uses the great circle distance to connect two vertices to construct a polygon area or line. If there is not enough density (that is, the number of points) for a set of vertices, then the line or the polygon area might be misinterpreted or the metadata might be considered invalid.

The ECHO system will not manipulate any of the spatial input metadata. You are responsible for the correctness and integrity of your spatial metadata. Prior to sending your metadata to ECHO for ingest, notify ECHO Operations as to which coordinate system (Cartesian or Geodetic) you used.

To prepare your metadata such that ECHO can support polar search and Geodetic search, observe the following:

- a. The spatial metadata must be stored in one of three possible types: point, line, or polygon. Choose an appropriate metadata type for your spatial metadata.
- b. Use only one coordinate system per data set.

- c. Provide spatial data with appropriate density if using the Geodetic model.
- d. Ingest for a metadata record will fail if any spatial metadata input is invalid under Oracle Spatial—refer back to Table 2 if needed.
- e. By default, ECHO assumes the maximum spatial coverage area for any Data Partner is the whole Earth (−180 to 180 for longitude and −90 to 90 for latitude). Specify the maximum spatial coverage area only if it varies from the default.
- f. Keep in mind the restrictions regarding lines and polygons described in Table 2.
- g. Provide the resolution (the default is 0.0001) for both latitude and longitude for your spatial metadata in advance. Your metadata generation application must define the resolution for vertices' degree representation. If any two vertices' difference is less than the resolution, those two vertices will be considered identical, which might cause the spatial metadata to be invalid for Oracle Spatial.

### 3.4.3 Temporal Data

Temporal metadata refer to a date and time associated with a collection and are essential search criteria for collections.

The ECHO 10.0 Format requires date & times to be formatted according the XML Schema Standard `dateTime` format

Your collection may be associated with one or more of the following three types of temporal expressions:  
`SingleDateTime`, `RangeDateTime` and `PeriodicDateTime`.

#### Code Listing 2. Expression of Temporal Information (RangeDateTime)

---

```
<Temporal>
  <DateType>Gregorian</DateType>
  <TemporalRangeType>Continuous Range</TemporalRangeType>
  <PrecisionofSeconds>1</PrecisionofSeconds>
  <EndsatPresentFlag>Y</EndsatPresentFlag>
  <RangeDateTime>
    <BeginningDateTime>1998-01-01T00:00:00.0Z</BeginningDateTime>
  </RangeDateTime>
</Temporal>
```

---

### 3.4.4 Sources and Sensors

ECHO adopts a layered representation of source and sensor information for the collection. A source and sensor layer is defined as:

---

```
Platform->* Instrument->* Sensor
```

---

A collection could be associated with zero (0) or more platforms; each platform could contain zero or more instruments, and each instrument could contain zero or more sensors. There might be characteristic parameters associated with the platforms, instruments, and/or sensors for the collection. There may also be an operational mode associated with instruments.

#### Code Listing 3. Full Platform/Instrument/Sensor Description

---

```
<Platform>
  <ShortName>Terra</ShortName>
  <LongName>
    First EOS Polar Orbiting Satellite, 10:30 AM Descending Equator Crossing
  </LongName>
  <Type>Spacecraft</Type>
```

---

```
<Characteristics>
  <Characteristic>

    <Name>EquatorCrossingTime</Name>
    <Description>
      Local time of the equator crossing and direction
      (ascending or descending)
    </Description>
    <DataType>varchar</DataType>
    <Unit>Local Mean Time</Unit>
    <Value>10:30, descending</Value>
  </Characteristic>
</Characteristics>
<Instruments>
  <Instrument>
    <ShortName>MODIS</ShortName>
    <LongName>
      Moderate-Resolution Imaging Spectroradiometer
    </LongName>
    <Technique>Imaging Spectroradiometry</Technique>
    <NumberOfSensors>2</NumberOfSensors>
    <Characteristics />
    <Sensors>
      <Sensor>
        <ShortName>MODIS</ShortName>
        <LongName>Cross-track Scanning Radiometer</LongName>
        <Technique>Radiometry</Technique>
        <Characteristics />
      </Sensor>
    </Sensors>
  </Instrument>
</Instruments>
</Platform>
```

---

### 3.4.5 Keywords

ECHO supports three kinds of keyword associations for collections: discipline keywords, spatial keywords, and temporal keywords. Discipline keywords and spatial keywords must comply with the Global Change Master Directory (GCMD) keywords standard located at <http://gcmd.gsfc.nasa.gov/Resources/valids/index.html>.

---

#### Code Listing 4. Sample Keywords

---

```
<SpatialKeywords>
  <Keyword>GEOGRAPHIC REGION</Keyword>
</SpatialKeywords>
<TemporalKeywords>
  <Keyword>UTC</Keyword>
</TemporalKeywords>
<DisciplineTopicParameters>
  <DisciplineTopicParameter>
    <DisciplineKeyword>EARTH SCIENCE</DisciplineKeyword>
    <TopicKeyword>ATMOSPHERE</TopicKeyword>
    <TermKeyword>ATMOSPHERIC CHEMISTRY</TermKeyword>
    <VariableKeyword>OXYGEN COMPOUNDS</VariableKeyword>
  </DisciplineTopicParameter>
```

```
</DisciplineTopicParameters>
```

### 3.4.6 Additional Attributes

Additional attributes are parameters, formerly known as Provider-Specific Attributes (PSAs), that further describe the collection and are important search criteria for the collection. ECHO supports collection-level additional attributes. Additional attributes may be used in a search. They may also be returned in a metadata query if specified.

#### Code Listing 5. Single Additional Attributes

```
<AdditionalAttributes>
  <AdditionalAttribute>
    <Name>
      VerParm_chlor_a_2
    </Name>
    <DataType>
      varchar
    </DataType>
    <Description>
      version of the parameter
    </Description>
  </AdditionalAttribute>
</AdditionalAttributes>
```

#### Code Listing 6. Collection-Level Additional Attributes

```
<AdditionalAttributes>
<AdditionalAttribute>
  <DataType>varchar</DataType>
  <Description>
    Version of the process software that generated the product.
  </Description>
  <Name>PROCESSVERSION</Name>
</AdditionalAttribute>
<AdditionalAttribute>
  <DataType>String</DataType>
  <Description>Estimated RMS error in geolocation</Description>
  <Name>GEO_EST_RMS_ERROR</Name>
</AdditionalAttribute>
<AdditionalAttribute>
  <DataType>String</DataType>
  <Description>Flag set (to 0) if science_state, the L1A engineering data flag
that indicates the Normal/Test configuration of the MODIS instrument, was set
for at least one scan in the granule.</Description>
  <Name>SCI_STATE</Name>
</AdditionalAttribute>
<AdditionalAttribute>
  <DataType>String</DataType>
  <Description>Flag set (to 0) if science_abnormal, the L1A engineering data
ground-set flag that indicates potentially abnormal science data due to things
other than MODIS (such as maneuvers, data link, etc.), was set for at least one
scan in the granule.</Description>
  <Name>SCI_ABNORM</Name>
</AdditionalAttribute>
```

```
<AdditionalAttribute>
  <DataType>int</DataType>

  <Description>The number of the granule for the day starting at midnight.
  Example: The granule from 00:00:00 to 00:00:05 will be granule number 1. The
  granule from 00:01:00 to 00:01:05 will be granule number 13</Description>

  <Name>GRANULENUMBER</Name>
</AdditionalAttribute>
</AdditionalAttributes>
```

---

### 3.4.7 Restriction Flag for a Collection

ECHO will restrict collections from being viewed and ordered by anyone except users bearing the provider role by verifying collection's Visibility Flag. By setting the visibility flag to be restricted before ingest of a new collection, ECHO will automatically restrict the new collection from viewing and ordering by all except the user bearing the Data Provider's role. The default restriction flag used for all granules in the collection can be overridden by setting restriction flag on a per granule basis.

Set a value (integer) in **RestrictionFlag** to indicate access constraints on a collection. ECHO references this value, in combination with your own data access rule, to restrict public access to the collection. You may use any range of integers for the RestrictionFlag value and assign your own meanings to these numbers.

- a. Granules inherit the value of their collection.
- b. If the collection has no value for the Restriction Flag, then there will be no default value for granules within the collection.

For example:

- You decide to use RestrictionFlag for a data quality summary, with a range of values from 0 to 10, with 0 indicating unknown quality, 1 indicating poor quality, and 10 indicating excellent quality,
- You have established a data access rule that restricts access to granules with a RestrictionFlag value of less than or equal to 5
- This means that Guest users will only be allowed to view granules with a RestrictionFlag value of 6 or higher.

Based on this example, the granule below will not be restricted from any user.

---

```
<RestrictionFlag>7</RestrictionFlag>
<RestrictionComment>default for collection</RestrictionComment>
```

---

For information about setting the Restriction Flag at the granule level, refer to section 3.5.8, Restriction Flag for a Granule.

## 3.5 ECHO GRANULES

The complete ECHO Granule format is on the Reference page of the ECHO website by selecting "10.0 Ingest Formats" under "Current System" (<http://www.echo.nasa.gov/reference/reference.shtml>).

### 3.5.1 Granule/Collection Association

ECHO's architecture is based on granules. Every granule must belong to a collection. A collection may contain zero or more granules. The granule/collection association information is provided via granule metadata input.

**Code Listing 7. Granule/Collection Association**

```
<GranuleMetaDataFile>
  <Granule>
    <GranuleUR>SC:MOD10A1.005:17363931</GranuleUR>
    <InsertTime>2007-09-09T00:35:41.283Z</InsertTime>

    <LastUpdate>2007-09-09T00:35:41.283Z</LastUpdate>

    <Collection>
      <ShortName>MOD10A1</ShortName>
      <VersionId>5</VersionId>
    </Collection>
    ...
  </Granule>
</GranuleMetaDataFile>
```

**Code Listing 8. Minimum Metadata Required To Insert A Granule**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<GranuleMetaDataFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Granule.xsd">
  <Granules>
    <Granule>
      <GranuleUR>ScAllRequired001</GranuleUR>
      <InsertTime>2002-12-12T00:54:36Z</InsertTime>
      <LastUpdate>2002-12-12T02:46:24Z</LastUpdate>
      <Collection>
        <ShortName>FOR_GRANULE_INSERT1</ShortName>
        <VersionId>1</VersionId>
      </Collection>
      <Orderable>true</Orderable>
    </Granule>
  </Granules>
</GranuleMetaDataFile>
```

Much information associated with the granule is defined or restricted by its collection. A collection's short name and version ID, combined, uniquely identify the collection to which the granule belongs. A granule can be associated with its collection by using the Data Set ID instead of short name and version number. When a collection is deleted, its granules will be deleted as well.

### 3.5.2 Granule Information

a. The required elements for granule metadata are:

- 1) GranuleUR
- 2) InsertTime
- 3) LastUpdate
- 4) Collection
- 5) Orderable

b. Additional (Optional) Elements

- 1) AdditionalAttributes
- 2) AssociatedBrowseImages

- 3) Campaigns
- 4) CloudCover
- 5) DataFormat
- 6) DataGranule
- 7) DeleteTime
- 8) InputGranules
- 9) MeasuredParameters
- 10) OnlineAccessURLs
- 11) OnlineResources
- 12) OrbitCalculatedSpatialDomains
- 13) PGEVersionClass
- 14) Platforms
- 15) Price
- 16) RestrictionComment
- 17) RestrictionFlag
- 18) Spatial
- 19) Temporal
- 20) TwoDCoordinateSystem
- 21) Visible

These elements can help describe your granule metadata more completely for the purpose of data searches.

- **RestrictionFlag** indicates whether access constraints should be applied when the granule data goes public. If the restriction setting is not explicitly indicated for a granule (via **RestrictionFlag**), it inherits the restriction setting of its associated collection.
- **InsertTime** is the day and time granule metadata was first inserted in your database.
- **LastUpdate** is the day and time the granule metadata was last updated in your database.

Your granule may be associated with **SingleDateTime** or **RangeDateTime**, representing the time when granule data were acquired. The granule's temporal information must be formatted according the XML Schema Standard `dateTime` format, and it must fall within the range of its collection's temporal system definition. Temporal information is an essential piece of metadata for a granule search. Since the temporal system definition is provided by its primary collection, in a granule's metadata, input the temporal system definition if it is not present.

#### **Code Listing 9. Granule Information**

---

```
<Granule>
  <GranuleUR>SC:MOD10A1.005:17363931</GranuleUR>
  <InsertTime>2007-09-09T00:35:41.283Z</InsertTime>
  <LastUpdate>2007-09-09T00:35:41.283Z</LastUpdate>
  <Collection>
    <ShortName>MOD10A1</ShortName>
    <VersionId>5</VersionId>
  </Collection>
  <DataGranule>
    <SizeMBDataGranule>0.487964</SizeMBDataGranule>
    <ReprocessingPlanned>
```

---

```
        further update is anticipated
    </ReprocessingPlanned>
    <ReprocessingActual>reprocessed</ReprocessingActual>
    <ProducerGranuleId>
        MOD10A1.A2002364.h13v12.005.2007251235931.hdf
    </ProducerGranuleId>
    <DayNightFlag>DAY</DayNightFlag>

    <ProductionDateTime>2007-09-08T20:01:58Z</ProductionDateTime>

    <LocalVersionId>SCF V5.0.5</LocalVersionId>
</DataGranule>
<PGEVersionClass>
    <PGEVersion>5.0.11</PGEVersion>
</PGEVersionClass>
<Temporal>
    <RangeDateTime>
        <BeginningDateTime>2002-12-30T08:35:00Z</BeginningDateTime>
        <EndingDateTime>2002-12-30T10:20:00Z</EndingDateTime>
    </RangeDateTime>
</Temporal>
<Orderable>true</Orderable>
</Granule>
```

---

### 3.5.3 Sources and Sensors

ECHO adopts a layered representation of source and sensor information for a spatial granule. A source and sensor layers are defined as:

---

```
Platform->* Instrument->* Sensor
```

---

A granule may be associated with zero or more platforms; each platform may contain zero or more instruments, and each instrument may contain zero or more sensors. In addition, a granule may define a sensor-level characteristic value for the characteristic parameters defined by its primary collection. In addition to the characteristic parameters, there might be an operational mode associated with instruments independent of its primary collection.

---

#### Code Listing 10. Sources and Sensors

---

```
<Platform>
    <ShortName>Terra</ShortName>
    <Instruments>
        <Instrument>
            <ShortName>MODIS</ShortName>
            <Sensors>
                <Sensor>
                    <ShortName>MODIS</ShortName>
                </Sensor>
            </Sensors>
        </Instrument>
    </Instruments>
</Platform>
```

---

The platform, instrument, and sensor referred to by the granule are restricted by its primary collection source/sensor reference and must comply with the GCMD standard, located at <http://gcmd.gsfc.nasa.gov/Resources/valids/index.html>.

### 3.5.4 Additional Attributes

Additional Attributes are parameters that further describe the granule and are important search criteria for the granule. The granule's Additional Attributes should fall within the Additional Attributes defined for its associated collection.

#### Code Listing 11. Granule level additional attributes

---

```
<AdditionalAttributes>
  <AdditionalAttribute>
    <Name>VERTICALTILENUMBER</Name>
    <Values>
      <Value>12</Value>
    </Values>
  </AdditionalAttribute>
  <AdditionalAttribute>
    <Name>TileID</Name>
    <Values>
      <Value>51013012</Value>
    </Values>
  </AdditionalAttribute>
  <AdditionalAttribute>
    <Name>QAPERCENTOTHERQUALITY</Name>
    <Values>
      <Value>2</Value>
    </Values>
  </AdditionalAttribute>
  <AdditionalAttribute>
    <Name>SNOWCOVERPERCENT</Name>
    <Values>
      <Value>0</Value>
    </Values>
  </AdditionalAttribute>
  <AdditionalAttribute>
    <Name>QAPERCENTGOODQUALITY</Name>
    <Values>
      <Value>98</Value>
    </Values>
  </AdditionalAttribute>
  <AdditionalAttribute>
    <Name>HORIZONTALTILENUMBER</Name>
    <Values>
      <Value>13</Value>
    </Values>
  </AdditionalAttribute>
</AdditionalAttributes>
```

---

#### Code Listing 12. ECHO 10 Additional Attributes

---

```
<AdditionalAttributes>
  <AdditionalAttribute>
```

---

```
<Name>VerParm_chlor_a_2</Name>
<DataType>varchar</DataType>
<Description>version of the parameter</Description>
</AdditionalAttribute>
</AdditionalAttributes>
```

---

### 3.5.5 Measured Parameters

Measured parameters are associated at the granule level only and are important search criteria for granules. For some providers, the value of certain measured parameters determines the visibility of the granule.

Measured parameters contain the name of the geophysical parameter expressed in the data as well as associated quality flags and quality status. The quality status contains measures of quality for the granule. The parameters used to set these measures are not preset and will be determined by the data producer. Each set of measures can occur many times either for the granule as a whole or for individual parameters. The quality flags contain the science, operational and automatic quality flags that indicate the overall quality assurance levels of specific parameter values within a granule.

#### Code Listing 13. Measured Parameters

---

```
<MeasuredParameters>
  <MeasuredParameter>
    <ParameterName>Snow_Cover_Daily_Tile</ParameterName>
    <QAStats>
      <QAPercentMissingData>0</QAPercentMissingData>
      <QAPercentCloudCover>65</QAPercentCloudCover>
    </QAStats>
    <QAFlags>
      <AutomaticQualityFlag>Passed</AutomaticQualityFlag>
      <AutomaticQualityFlagExplanation>
        No automatic quality assessment done in the PGE
      </AutomaticQualityFlagExplanation>
      <OperationalQualityFlag>Passed</OperationalQualityFlag>
      <OperationalQualityFlagExplanation>
        Passed
      </OperationalQualityFlagExplanation>
      <ScienceQualityFlag>Not Investigated</ScienceQualityFlag>
      <ScienceQualityFlagExplanation>
        See
        http://landweb.nascom.nasa.gov/cgi-
bin/QA_WWW/qaFlagPage.cgi?sat=terra
        for the product Science Quality status.
      </ScienceQualityFlagExplanation>
    </QAFlags>
  </MeasuredParameter>
</MeasuredParameters>
```

---

### 3.5.6 Orbital Information

The following code sample shows how to include information associated with a satellite's orbit.

*Note: orbit granules may not be ingested unless the parent collections have orbit parameters defined*

For more information on orbit data, see section 3.13, Orbit Data.

---

#### Code Listing 14. Orbital Information

---

```
<OrbitCalculatedSpatialDomains>
  <OrbitCalculatedSpatialDomain>
    <OrbitalModelName>OrbitalModelName1</OrbitalModelName>
    <OrbitNumber>7694</OrbitNumber>
    <StartOrbitNumber>0.0</StartOrbitNumber>
    <StopOrbitNumber>0.0</StopOrbitNumber>
    <EquatorCrossingLongitude>0.0</EquatorCrossingLongitude>

    <EquatorCrossingDateTime>2009-01-05T05:30:30-
  05:00</EquatorCrossingDateTime>

  </OrbitCalculatedSpatialDomain>
</OrbitCalculatedSpatialDomains>
```

---

### 3.5.7 Granule/Browse Association

A granule can be associated with zero (0) or more browse files, and a browse file can be referenced by more than one granule. Browse associations are defined using by the ProviderBrowseId of a browse file that has been ingest in ECHO. For more information on browse metadata ingest, refer to section 3.9 Browse Images and Browse Image Metadata.

The following code sample shows how to include a granule/browse association:

---

#### Code Listing 15. Orbital Information

---

```
<AssociatedBrowseImages>
  <ProviderBrowseId>BR:19285</ProviderBrowseId>
</AssociatedBrowseImages>
```

---

### 3.5.8 Restriction Flag for a Granule

Set a value (integer) in **RestrictionFlag** to indicate access constraints on a granule. ECHO references this value, in combination with your own data access rule, to restrict public access to the granule. You may use any range of integers for the RestrictionFlag value and assign your own meanings to these numbers.

- If you indicate a Restriction Flag setting for a granule, it will override the default value for its collection.
- If you do not indicate a Restriction Flag setting for a granule, it will inherit the value of its collection.
- If the collection has no value for the Restriction Flag, then there will be no default value for granules within the collection.

For example:

- You decide to use RestrictionFlag for a data quality summary, with a range of values from 0 to 10, with 0 indicating unknown quality, 1 indicating poor quality, and 10 indicating excellent quality
- You have established a data access rule that restricts access to granules with a RestrictionFlag value of less than or equal to 5
- This means Guest users will only be allowed to view granules with a RestrictionFlag value of 6 or higher.

Based on this example, the granule below will not be restricted from any user.

#### Code Listing 16. Restriction Flag Set for a Granule

```
<RestrictionFlag>8</RestrictionFlag>
<RestrictionComment>quality summary value of 8 (good)</RestrictionComment>
```

*Note: Restriction flag set for a granule overrides the default Collection value.*

### 3.5.9 Spatial Representations, Coordinates and Projections

For a collection or granule, the spatial area coverage is one of the most important and basic search criteria for Earth science data, although it is not required. To submit the metadata so that spatial area coverage can be used as part of a basic search, apply the concepts discussed throughout section 3.5, ECHO Granules.

ECHO supports multiple spatial representations. Each provider can have different spatial representation; however, each collection must define a single spatial representation for its granules.

ECHO supports three coordinate systems for spatial data. Each Data Partner should use only one coordinate system when constructing the spatial area coverage for a collection or granule.

### 3.5.10 Two-Dimensional Coordinate System

Granule-specific values can be searched against a two-dimensional coordinate system-based catalog search for Coordinate1 and Coordinate2. This search ability is not supported by all Data Providers.

Examples of two-dimensional coordinate system values are path/row for Worldwide Reference System (WRS) data and Moderate Resolution Imaging Spectroradiometer (MODIS) tiles IDs.

*Note: Two-dimensional coordinate system types are defined by ECHO operations and have to be enabled for a collection before it can be used in granule ingest.*

#### Code Listing 17. Setting Two-Dimensional Coordinate System Coordinates

```
<TwoDCoordinateSystem>
<StartCoordinate1>21</StartCoordinate1>
<StartCoordinate2>29</StartCoordinate2>
<EndCoordinate2>33</EndCoordinate2>
<TwoDCoordinateSystemName>WRS2</TwoDCoordinateSystemName>
</TwoDCoordinateSystem>
```

## 3.6 GEOMETRY REPRESENTATIONS

Spatial data are most commonly described as geometry such as a polygon, a multi-polygon, or a line. They are stored as Oracle spatial objects in the database to record shape, spatial locations of corner points and spatial coordinate system used, Cartesian or Geodetic. Cartesian and Geodetic are considered different spatial representations. Corner points in the Cartesian system are connected by straight lines, while in the Geodetic system they are connected by great circle arcs. This makes their spatial coverage slightly different. Oracle requires specifying the coordinate system during data storage time. Therefore, data under different coordinate systems are also searched differently.

## 3.6.1 Coordinate System

### 3.6.1.1 Cartesian Coordinate System

The Cartesian coordinate system is a flattened coordinate system with longitude ranged from –180 to 180 degrees and latitude ranged from –90 to 90 degrees. The projected map is flattened and open along the International Date Line with North Pole and South Pole as top and bottom line respectively.

### 3.6.1.2 Geodetic Coordinate System

The Geodetic coordinate system is defined in angular (latitude and longitude) and is defined relative to spherical polar coordinate and Earth Geodetic datum. Oracle defines coordinate system following OGC standards, which are defined at <http://www.opengeospatial.org/>. The Geodetic coordinate ECHO chose to support is World Geodetic System 84 (WGS 84)—refer to section 3.1.2, Metadata Mapping/Ingest Process Considerations, for the previous discussion of Geodetic coordinate system.

Also, refer to Table 2 for supported spatial data types.

The following code sample shows how to define Geodetic coordinate system:

---

```
GEOGCS [ "Longitude / Latitude (WGS 84)", DATUM ["WGS 84", SPHEROID ["WGS 84",
6378137.000000,
298.257224]], PRIMEM [ "Greenwich", 0.000000 ], UNIT ["Decimal Degree",
0.01745329251994330]]
```

---

## 3.6.2 Data Types and Representation

### 3.6.2.1 Geometry

Spatial data in Cartesian or Geodetic coordinate systems are specified within a <Geometry> tag.

#### Code Listing 18. Geometry Example

---

```
<Spatial>
  <HorizontalSpatialDomainContainer>
    <Geometry>
      ...
    </HorizontalSpatialDomainContainer>
  </Spatial>
```

---

Within a <Geometry> tag points, lines, etc. can be included to define the spatial extent of your data.

### 3.6.2.2 Point

ECHO can receive and store spatial data representing one or more points. ECHO also supports searching for spatial data representing one or more points. In the XML metadata, follow the syntax shown in the following code sample to define a granule's spatial extent as a point:

#### Code Listing 19. Single Point Example

---

```
<Geometry>
  <Point>
    <PointLongitude>-123.948</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
  </Point>
</Geometry>
```

---

---

**Code Listing 20. Multiple Points**

---

```
<Geometry>
  <Point>
    <PointLongitude>-123.948</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
  </Point>
  <Point>
    <PointLongitude>-133.546</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
  </Point>
</Geometry>
```

---

**3.6.2.3 Line**

ECHO can receive and store spatial data representing one or more lines. ECHO also supports searching for spatial data representing one or more lines. In the XML metadata, follow the syntax shown in the following code sample to define a granule's spatial extent as a line:

---

**Code Listing 21. Single Line Example**

---

```
<Geometry>
  <Line>
    <Point>
      <PointLongitude>-123.948</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-133.546</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
  </Line>
</Geometry>
```

---

---

**Code Listing 22. Multiple Line Example**

---

```
<Geometry>
  <Line>
    <Point>
      <PointLongitude>-123.948</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-133.546</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
  </Line>
  <Line>
```

---

```
<Point>
    <PointLongitude>-123.948</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
</Point>
<Point>
    <PointLongitude>-133.546</PointLongitude>
    <PointLatitude>45.0664</PointLatitude>
</Point>
<Point>
    <PointLongitude>-143.546</PointLongitude>
    <PointLatitude>40.0664</PointLatitude>
</Point>
</Line>
</Geometry>
```

---

In the Cartesian coordinate system, the points are connected with a straight line on the plane in the order listed. The line connects two points and will never cross the International Date Line or Poles.

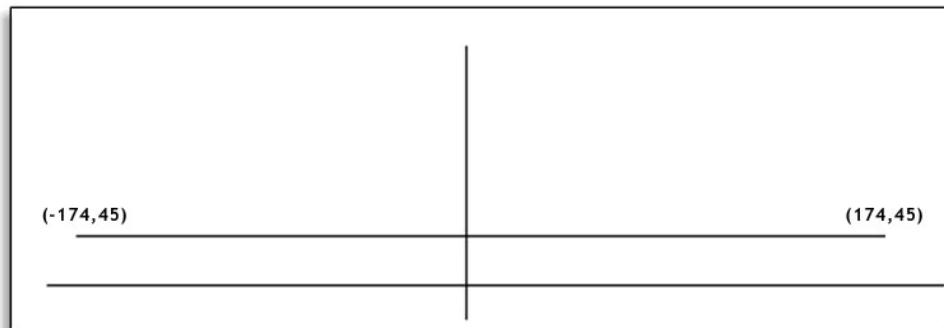
#### **Code Listing 23. Interpreted in Cartesian and Geodetic Systems**

---

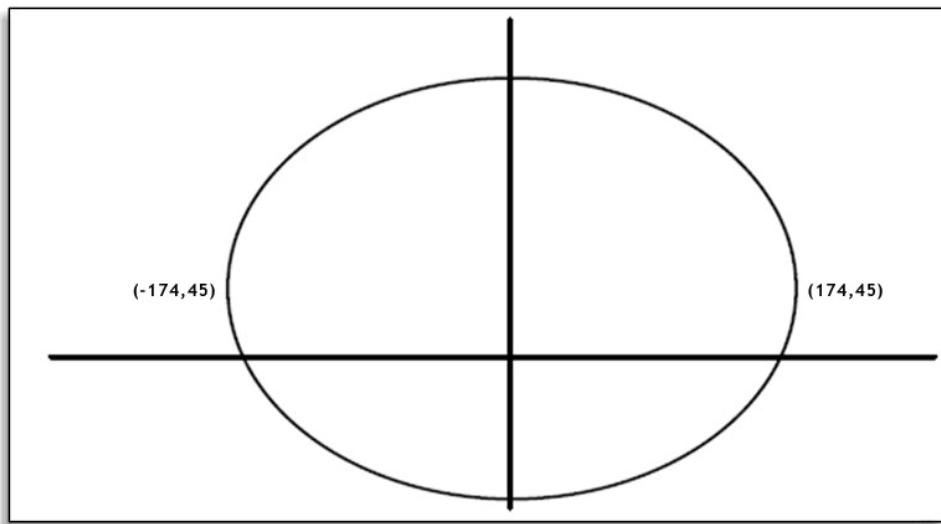
```
<Geometry>
<Line>
    <Point>
        <PointLongitude>-173.948</PointLongitude>
        <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
        <PointLongitude>173.546</PointLongitude>
        <PointLatitude>45.0664</PointLatitude>
    </Point>
</Line>
</Geometry>
```

---

The figure below shows a line in the Cartesian coordinate system:



This same expression in the Geodetic coordinate system represents the line as:



In the Geodetic coordinate system, the line connects two points using a great circle arc with the shortest distance between the two points. The line could cross the International Date Line and poles. If the same line indicated in the Cartesian coordinate system is represented in the Geodetic coordinate system, the code expression would be:

#### Code Listing 24. Adding Density

```
<Geometry>
  <Line>
    <Point>
      <PointLongitude>-173.948</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>0.0</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>173.546</PointLongitude>
      <PointLatitude>45.0664</PointLatitude>
    </Point>
  </Line>
</Geometry>
```

The additional points give the line more density so that Oracle interprets the data correctly. The appropriate density should be applied to the Geodetic coordinate system as well.

#### 3.6.2.4 Polygon

ECHO can receive and store spatial data representing a polygon, a polygon with hole, and multiple polygons (with or without holes). ECHO also supports searching for spatial data representing any of these polygons. In the XML metadata, follow the syntax shown in the following code sample to define a granule's spatial extent as a polygon:

**Code Listing 25. Single Polygon**

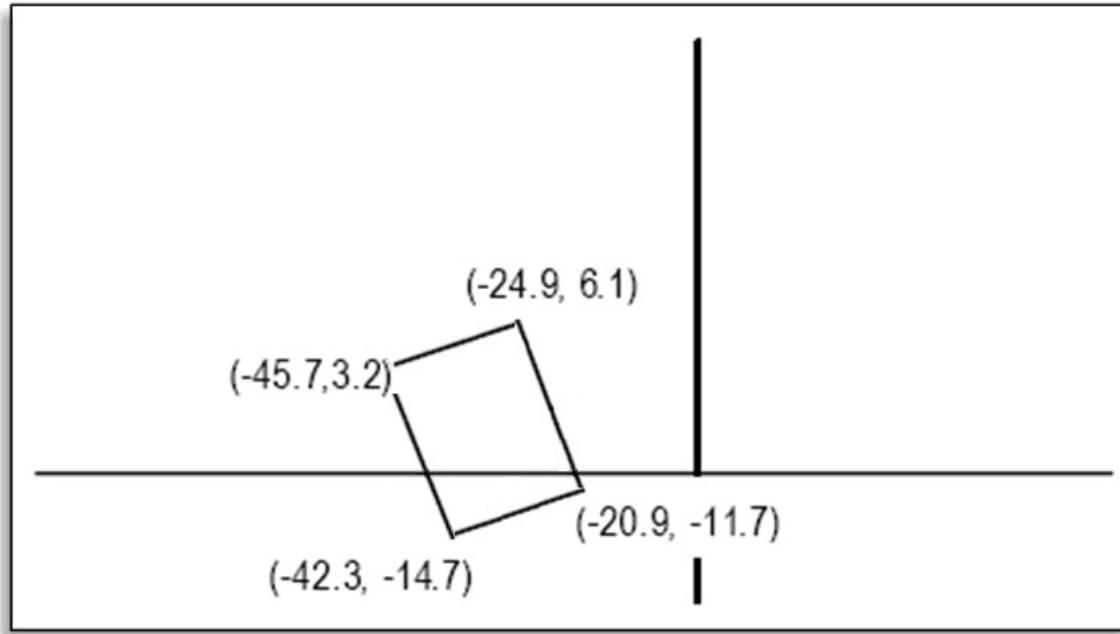
---

```
<Geometry>
  <Polygon>
    <SinglePolygon>
      <OutRing>
        <Boundary>
          <Point>
            <PointLongitude>-20.9342</PointLongitude>
            <PointLatitude>-11.7045</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-42.3067</PointLongitude>
            <PointLatitude>-14.7732</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-45.7985</PointLongitude>
            <PointLatitude>3.198</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-24.8982</PointLongitude>
            <PointLatitude>6.1665</PointLatitude>
          </Point>
        </Boundary>
      </OutRing>
    </SinglePolygon>
  </Polygon>
</Geometry>
```

---

A single polygon can have multiple holes, each represented by a single outer ring surrounding the area within it. In the Cartesian coordinate system, straight lines connect the points of the ring in the order in which they are listed, which must always be in clockwise order. The area should not cross the International Date Line or the poles. In the Geodetic coordinate system, the points are connected using a great circle arc according to the shortest distance between two points. Remember that a polygon coverage cannot span more than half the earth and may not cross the International Date Line and/or poles in the Geodetic coordinate system.

The figure below uses the Cartesian coordinate system to represent the spatial area covered when applying the code above:



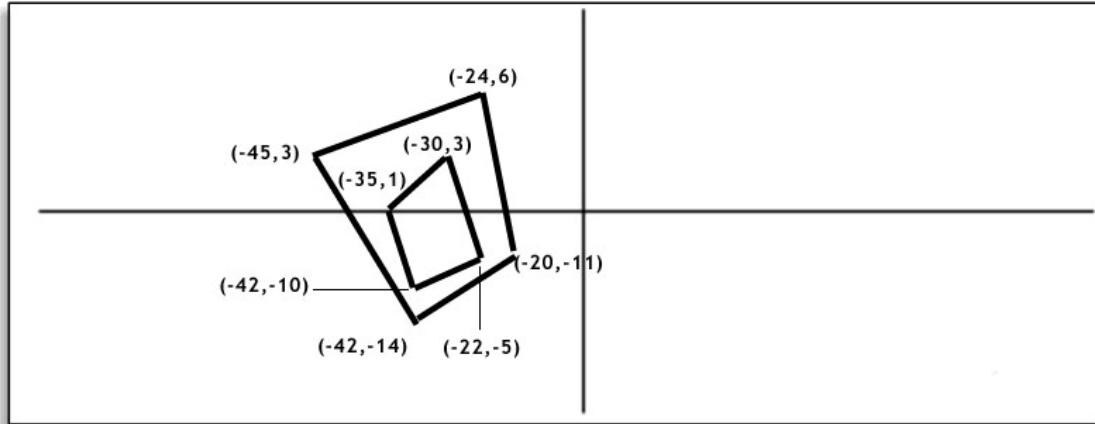
**Code Listing 26. Single Polygon with a Hole**

```
<Geometry>
  <Polygon>
    <SinglePolygon>
      <OutRing>
        <Boundary>
          <Point>
            <PointLongitude>-19.9342</PointLongitude>
            <PointLatitude>-10.7045</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-42.3067</PointLongitude>
            <PointLatitude>-13.7732</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-44.7985</PointLongitude>
            <PointLatitude>3.198</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-23.8982</PointLongitude>
            <PointLatitude>6.1665</PointLatitude>
          </Point>
        </Boundary>
      </OutRing>
      <InnerRing>
        <Boundary>
          <Point>
            <PointLongitude>-21.9342</PointLongitude>
```

```
<PointLatitude>-5.7045</PointLatitude>
</Point>
<Point>
    <PointLongitude>-42.3067</PointLongitude>
    <PointLatitude>-9.7732</PointLatitude>
</Point>
<Point>
    <PointLongitude>-34.7985</PointLongitude>
    <PointLatitude>1.198</PointLatitude>
</Point>
<Point>
    <PointLongitude>-29.8982</PointLongitude>
    <PointLatitude>3.1665</PointLatitude>
</Point>
</Boundary>
</InnerRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

While a single polygon with a hole can have only one outer ring that represents the area surrounded within, it can have multiple inner rings that represent holes. All the rules, restrictions and discussions for the outer ring in both coordinate systems apply to inner rings as well. An inner ring should be completely contained within the outer ring.

The figure below represents the spatial area covered when applying the code above:



### 3.6.2.5 Multiple Polygon

A multiple polygon is a combination of polygons that may or may not have holes. Each single polygon expression should follow the same rules listed for polygon above. No two polygons may overlap each other.

### 3.6.2.6 Bounding Box

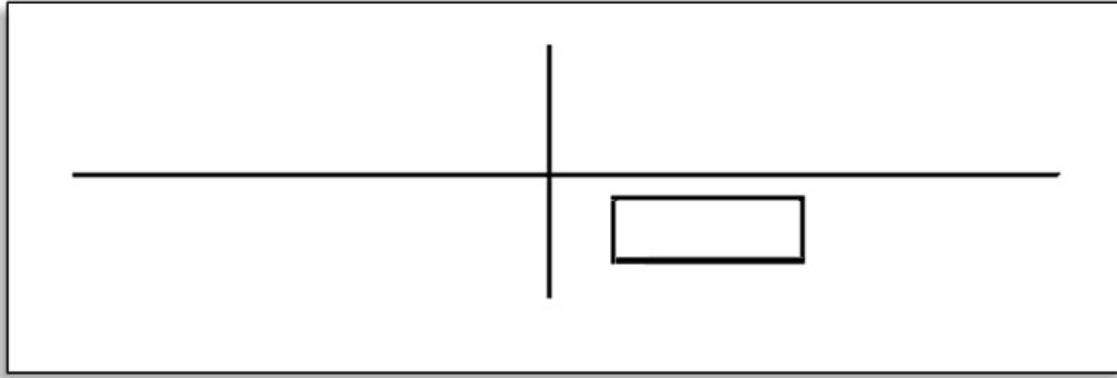
In the Cartesian coordinate system only, ECHO is capable of receiving, storing and supporting the search on spatial data representing a bounding box or multiple bounding boxes. To send ECHO spatial point data in the metadata XML file, follow the syntax shown in this code sample:

#### Code Listing 27. Bounding Box

```
<Geometry>
```

```
<BoundingRectangle>
  <WestBoundingCoordinate>8.733</WestBoundingCoordinate>
  <NorthBoundingCoordinate>-7.4861</NorthBoundingCoordinate>
  <EastBoundingCoordinate>43.199501</EastBoundingCoordinate>
  <SouthBoundingCoordinate>-35.2617</SouthBoundingCoordinate>
</BoundingRectangle>
</Geometry>
```

The figure below represents the spatial area covered when applying the code shown above:



ECHO stores a bounding box as a four-pointed polygon, subject to the specifications and constraints described for the polygon. Bounding boxes cannot encompass more than half the earth and cannot cross the International Date Line or the poles. If there is more than one bounding box listed, then ECHO stores them as multiple polygons but requires that they not overlap.

## 3.7 INVALID SPATIAL REPRESENTATION

### 3.7.1 Polygon Points in Counter-Clockwise Order

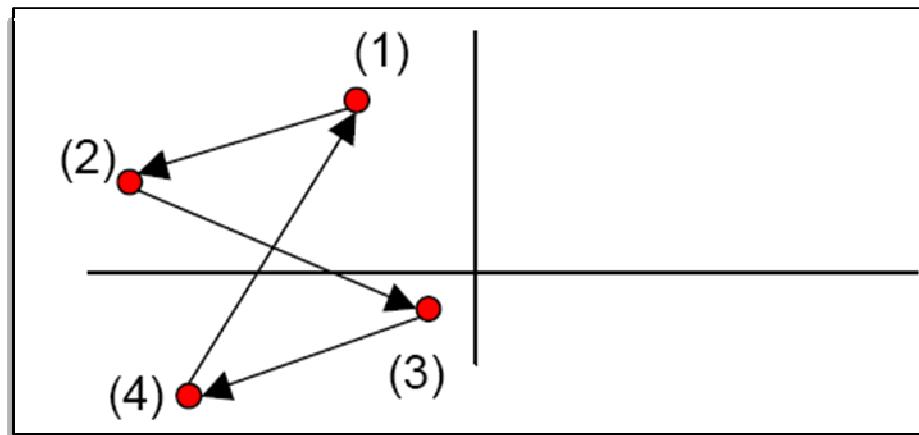
This spatial area expression is invalid in the Cartesian coordinate system. However, this same expression is considered valid in the Geodetic coordinate system, although the coverage will be interpreted differently.

#### Code Listing 28. Polygon with Points in Counter-Clockwise Order

```
<Geometry>
  <Polygon>
    <SinglePolygon>
      <OutRing>
        <Boundary>
          <Point>
            <PointLongitude>170</PointLongitude>
            <PointLatitude>30</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-170</PointLongitude>
            <PointLatitude>30</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-170</PointLongitude>
            <PointLatitude>-30</PointLatitude>
          </Point>
        </Boundary>
      </OutRing>
    </SinglePolygon>
  </Polygon>
</Geometry>
```

```
</Point>
<Point>
  <PointLongitude>170</PointLongitude>
  <PointLatitude>-30</PointLatitude>
</Point>
</Boundary>
</OutRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

### 3.7.2 Twisted Polygon



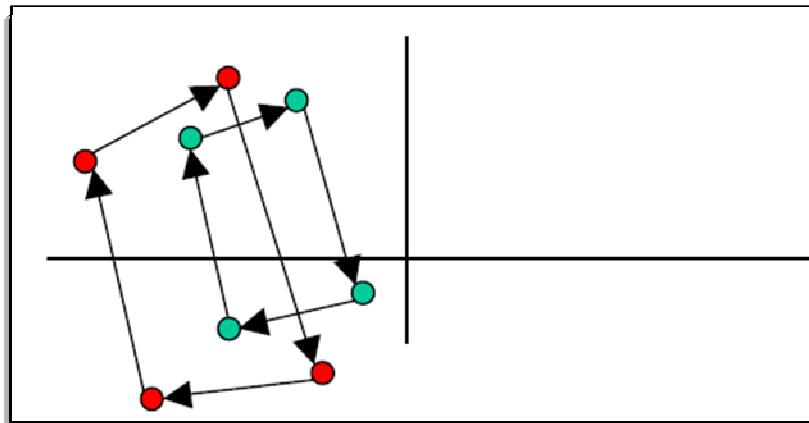
In either the Cartesian or the Geodetic coordinate systems, the following code sample will result in an invalid polygon:

**Code Listing 29. Twisted Polygon**

```
<Geometry>
<Polygon>
  <SinglePolygon>
    <OutRing>
      <Boundary>
        <Point>
          <PointLongitude>-20.9342</PointLongitude>
          <PointLatitude>-11.7045</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-42.3067</PointLongitude>
          <PointLatitude>-14.7732</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-24.8982</PointLongitude>
          <PointLatitude>6.1665</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-45.7985</PointLongitude>
```

```
<PointLatitude>3.198</PointLatitude>
</Point>
</Boundary>
</OutRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

### 3.7.3 Hole Crosses over Outer Ring



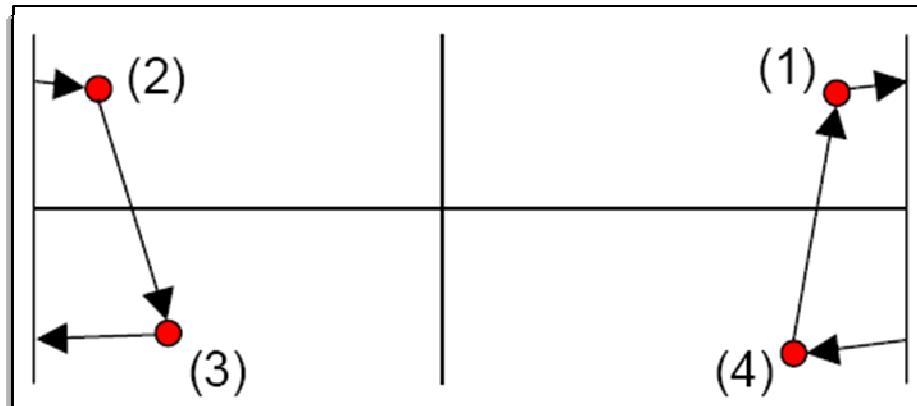
In either the Cartesian or the Geodetic coordinate systems, the following code sample will result in an invalid data:

**Code Listing 30. Hole Crosses over the Outer Ring**

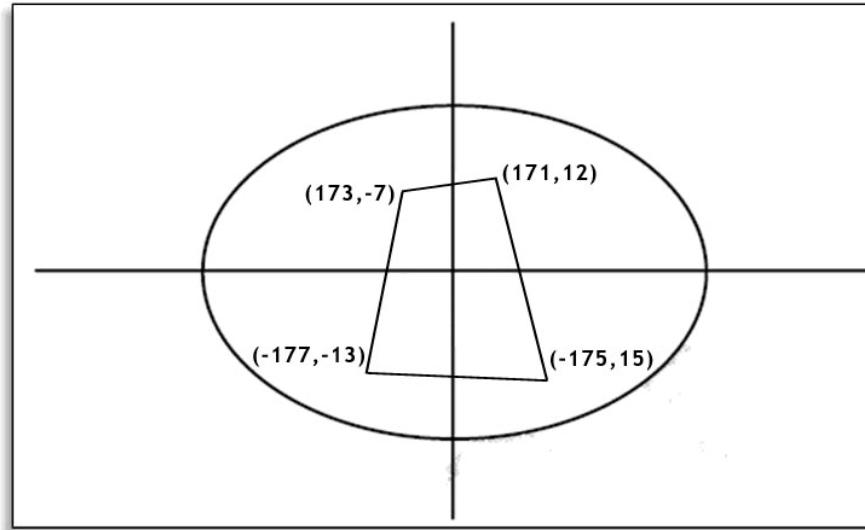
```
<Geometry>
<Polygon>
<SinglePolygon>
<OutRing>
<Boundary>
<Point>
<PointLongitude>-20.9342</PointLongitude>
<PointLatitude>-11.7045</PointLatitude>
</Point>
<Point>
<PointLongitude>-42.3067</PointLongitude>
<PointLatitude>-14.7732</PointLatitude>
</Point>
<Point>
<PointLongitude>-45.7985</PointLongitude>
<PointLatitude>3.198</PointLatitude>
</Point>
<Point>
<PointLongitude>-24.8982</PointLongitude>
<PointLatitude>6.1665</PointLatitude>
</Point>
</Boundary>
</OutRing>
```

```
<InnerRing>
  <Boundary>
    <Point>
      <PointLongitude>-17.9342</PointLongitude>
      <PointLatitude>-5.7045</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-30.3067</PointLongitude>
      <PointLatitude>-10.7732</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-35.7985</PointLongitude>
      <PointLatitude>1.198</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-10.8982</PointLongitude>
      <PointLatitude>3.1665</PointLatitude>
    </Point>
  </Boundary>
</InnerRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

### 3.7.4 Polygon Crosses International Date Line



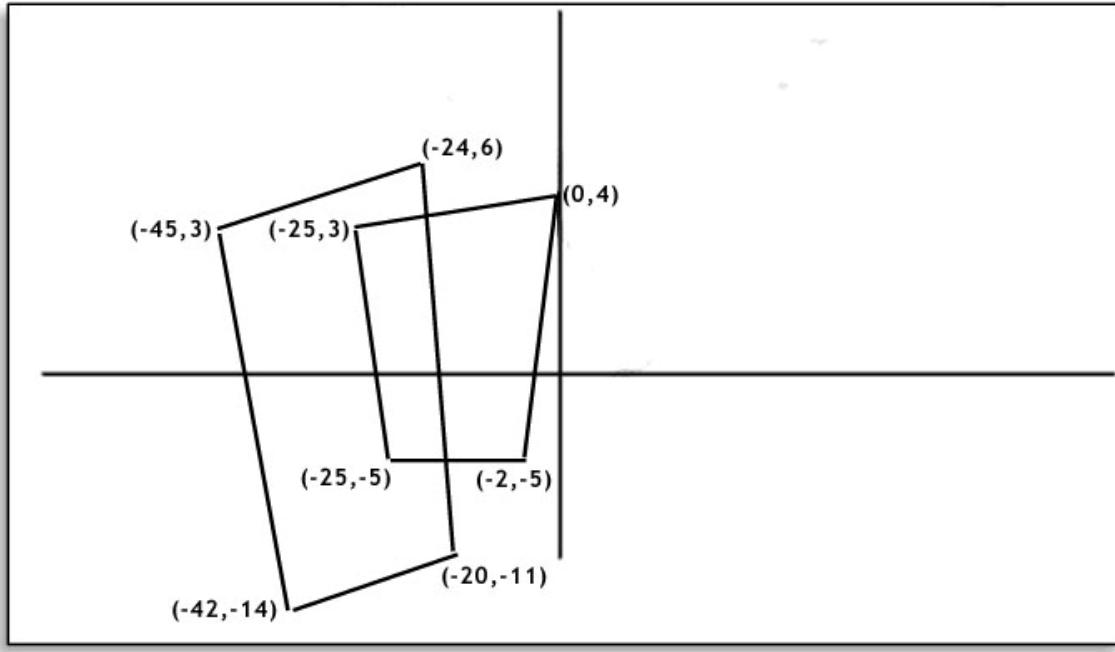
According to clockwise order, this expression in the Cartesian coordinate system represents a polygon crossing the International Date Line; therefore, it is invalid. However, in the Geodetic coordinate system, this is a valid spatial coverage area, as shown below:



**Code Listing 31. Polygon Crosses International Dateline**

```
<Geometry>
  <Polygon>
    <SinglePolygon>
      <OutRing>
        <Boundary>
          <Point>
            <PointLongitude>170.9342</PointLongitude>
            <PointLatitude>11.7045</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-175.3067</PointLongitude>
            <PointLatitude>14.7732</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-176.7985</PointLongitude>
            <PointLatitude>-13.198</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>172.8982</PointLongitude>
            <PointLatitude>-7.1665</PointLatitude>
          </Point>
        </Boundary>
      </OutRing>
    </SinglePolygon>
  </Polygon>
</Geometry>
```

### 3.7.5 Overlapping Polygons



In either coordinate system, this represents invalid spatial data.

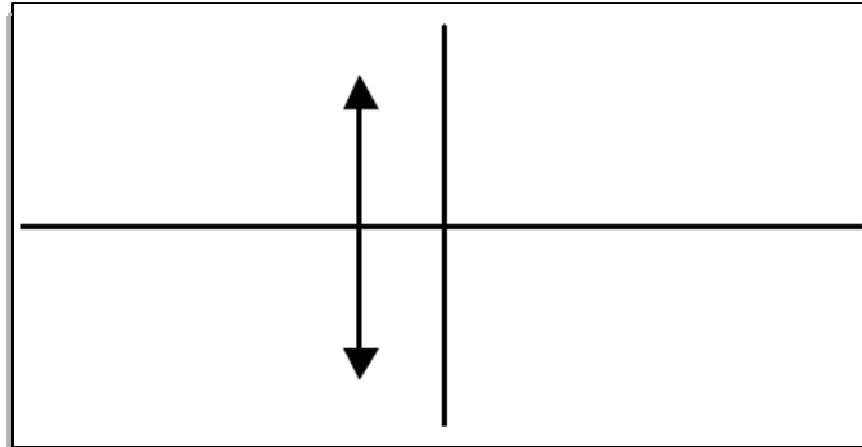
**Code Listing 32. Overlapping Polygons**

```
<Geometry>
  <Polygon>
    <SinglePolygon>
      <OutRing>
        <Boundary>
          <Point>
            <PointLongitude>-19.9342</PointLongitude>
            <PointLatitude>-10.7045</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-42.3067</PointLongitude>
            <PointLatitude>-13.7732</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-44.7985</PointLongitude>
            <PointLatitude>3.198</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-23.8982</PointLongitude>
            <PointLatitude>6.1665</PointLatitude>
          </Point>
        </Boundary>
      </OutRing>
    </SinglePolygon>
  </Polygon>
  <Polygon>
    <SinglePolygon>
```

```
<OutRing>
  <Boundary>

    <Point>
      <PointLongitude>-1.9342</PointLongitude>
      <PointLatitude>-4.7045</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-25.3067</PointLongitude>
      <PointLatitude>-4.7732</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-24.7985</PointLongitude>
      <PointLatitude>3.198</PointLatitude>
    </Point>
    <Point>
      <PointLongitude>-0.2982</PointLongitude>
      <PointLatitude>4.1665</PointLatitude>
    </Point>
  </Boundary>
</OutRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

### 3.7.6 Inappropriate Data



In this example, the bounding box is actually a line. It is important to use the correct spatial data type when constructing a spatial data expression. Do not use a bounding box to express a line.

*Note: The granularity of the spatial index for a given provider can result in two distinct but nearby points being interpreted as the same point.*

#### Code Listing 33. Inappropriate Data

```
<Geometry>
  <BoundingRectangle>
```

```
<WestBoundingCoordinate>-8.733</WestBoundingCoordinate>
<NorthBoundingCoordinate>7.4861</NorthBoundingCoordinate>

<EastBoundingCoordinate>-8.733</EastBoundingCoordinate>

<SouthBoundingCoordinate>10.2617</SouthBoundingCoordinate>
</BoundingRectangle>
</Geometry>
```

---

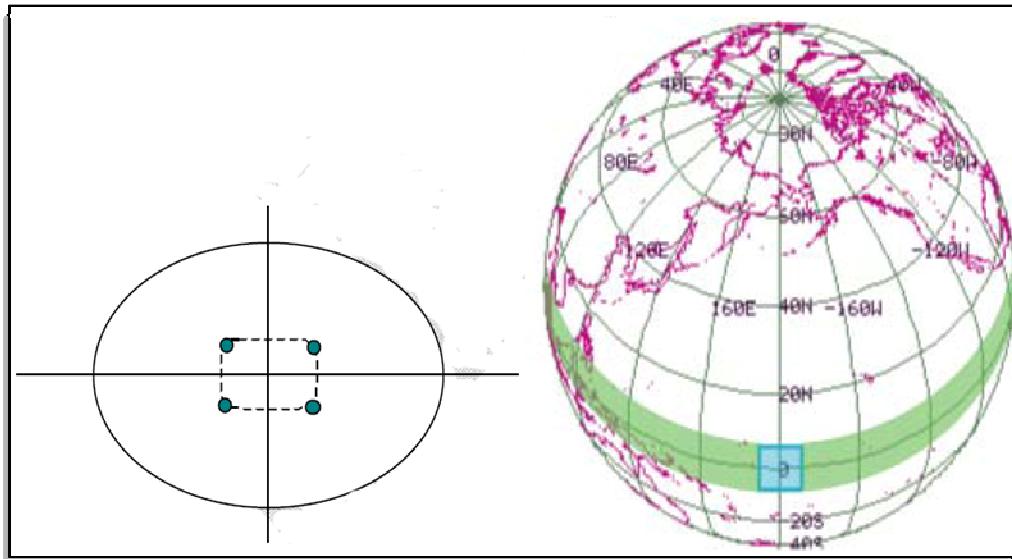
**Code Listing 34. Incorrect Density**

---

```
<Geometry>
<Polygon>
<SinglePolygon>
<OutRing>
<Boundary>
<Point>
    <PointLongitude>170.9342</PointLongitude>
    <PointLatitude>11.7045</PointLatitude>
</Point>
<Point>
    <PointLongitude>-175.3067</PointLongitude>
    <PointLatitude>14.7732</PointLatitude>
</Point>
<Point>
    <PointLongitude>-176.7985</PointLongitude>
    <PointLatitude>-13.198</PointLatitude>
</Point>
<Point>
    <PointLongitude>172.8982</PointLongitude>
    <PointLatitude>-7.1665</PointLatitude>
</Point>
</Boundary>
</OutRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

---

The expression above is valid spatial data in the Geodetic coordinate system. However, the spatial coverage area represented will be as shown below:



You may want the large green band, but you will receive the small blue rectangle.

Oracle connects any two points using the shortest distance between points. To represent this spatial coverage correctly, you must increase the points' density by adding extra points. The sample below shows one way you might express these additional points, to represent this spatial coverage area correctly.

#### Code Listing 35. Correct Density

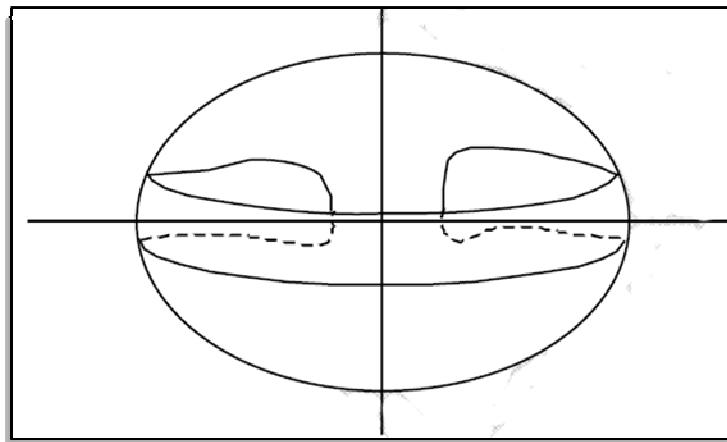
```
<Geometry>
  <Polygon>
    <SinglePolygon>
      <OutRing>
        <Boundary>
          <Point>
            <PointLongitude>170.9342</PointLongitude>
            <PointLatitude>11.7045</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>0.0</PointLongitude>
            <PointLatitude>14.7732</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-175.3067</PointLongitude>
            <PointLatitude>14.7732</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>-176.7985</PointLongitude>
            <PointLatitude>-13.198</PointLatitude>
          </Point>
          <Point>
            <PointLongitude>0.0</PointLongitude>
            <PointLatitude>-10.1665</PointLatitude>
          </Point>
          <Point>
```

```
<PointLongitude>172.8982</PointLongitude>
<PointLatitude>-7.1665</PointLatitude>

</Point>
</Boundary>
</OutRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

### 3.7.7 Incorrectly Defined Spatial Coverage

The spatial area coverage in this case is greater than half of the earth. This is invalid spatial data in both the Geodetic and the Cartesian coordinate systems because Oracle does not support spatial area greater than half of the earth in those two systems.



**Code Listing 36. Polygon Covering More Than Half of the Earth**

```
<Geometry>
<Polygon>
  <SinglePolygon>
    <OutRing>
      <Boundary>
        <Point>
          <PointLongitude>170.9342</PointLongitude>
          <PointLatitude>71.7045</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>0.0</PointLongitude>
          <PointLatitude>71.7732</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-175.3067</PointLongitude>
          <PointLatitude>71.7732</PointLatitude>
        </Point>
        <Point>
          <PointLongitude>-176.7985</PointLongitude>
          <PointLatitude>-71.198</PointLatitude>
        </Point>
      </Boundary>
    </SinglePolygon>
  </Polygon>
</Geometry>
```

```
<Point>
    <PointLongitude>0.0</PointLongitude>

    <PointLatitude>-71.1665</PointLatitude>
</Point>
<Point>
    <PointLongitude>172.8982</PointLongitude>
    <PointLatitude>-71.1665</PointLatitude>
</Point>
</Boundary>
</OutRing>
</SinglePolygon>
</Polygon>
</Geometry>
```

---

### 3.7.8 Latitude/Longitude Range and Tolerance

ECHO makes use of resolution and range parameter settings to associate a level of precision and data range with spatial data. ECHO uses these parameters as evaluation parameters when validating spatial data input. Spatial data that use latitude/longitude are expressed in degrees, and tolerance is expressed in meters. For example, if the range for latitude is -50 to 50 and the range for longitude is -150 to 150, then ECHO will consider any granule input data with latitude and/or longitude falling outside of this range to be invalid. If the tolerance is 0.05 for both latitude and longitude, and if the distance between two points is less than 0.05 meter for both longitude and latitude, then those two points are considered the same point. In this situation, the spatial expression is invalid because ECHO spatial constructs require each point to have a unique spatial location. You should specifically define latitude/longitude range within your collection metadata.

ECHO defaults for range and tolerance are:

- Resolution: .1 meter (.0001 Km)
- Tolerance: .05 meter

## 3.8 ONLINE DATA ACCESS URL AND ONLINE RESOURCES URL

For some granules or collections, the raw data are made available online via FTP or web URL. ECHO stores this online access information for directly accessible granule and collection data differently from information covering other aspects of granule and collection data. Directly accessible data require the `<OnlineAccessURLs>` tag and include the URL to that data. Use `OnlineAccessURLs` only for the actual data.

Any other online information covering aspects of the data, such as guides, product listings, validation information, etc., should be listed in a `<OnlineResources>`, along with the URLs to that information.

## 3.9 BROWSE IMAGES AND BROWSE IMAGE METADATA

The complete ECHO Browse Image format is available on the Reference page of the ECHO website by selecting "10.0 Ingest Formats" under "Current System" (<http://www.echo.nasa.gov/reference/reference.shtml>).

Browse images provide a high-level view of granule or collection data. If you have browse images, you should send both browse image files and a browse image metadata XML files to the ECHO system. The ECHO system will allocate the storage for browse image files, build a browse image URL, and associate the browse image URL to its item record in the ECHO database.

When ECHO processes a browse image ingest, all the browse image files must also be included. In addition to checking for the existence of the browse image files, ECHO also verifies the actual browse image file size against the file size indicated for that file in the browse image metadata input XML file. If any browse image file indicated

in the browse image metadata input XML file does not exist, or if any file size does not match the indicated size, then the browse image files and browse image metadata input XML file will not be processed.

ECHO supports many-to-many referencing between collections and browse images or granules and browse images. After processing, ECHO places the browse image files online. Using the Data Partner's unique identifier of the browse image, associations can then be made to collections or granules. This information is then made available to the end user.

---

#### Code Listing 37. Sample Browse Insert

---

```
<?xml version="1.0" encoding="UTF-8"?>
<BrowseMetaDataFile>
  <BrowseImages>
    <BrowseImage>
      <ProviderBrowseId>BR:036015124</ProviderBrowseId>
      <FileName>
        MOBCTT.A2004351.0425.005.2007036015124.jpg
      </FileName>
      <FileSize>558403</FileSize>
    </BrowseImage>
  </BrowseImages>
</BrowseMetaDataFile>
```

---

ECHO Ingest will load the sample browse and insert into the ECHO database, where it will remain until a browse image delete is submitted.

---

#### Code Listing 38. Sample Granule Insert with Browse Associations

---

```
<?xml version="1.0" encoding="UTF-8"?>
<GranuleMetaDataFile>
  <Granules>
    <Granule>
      <GranuleUR>GR:2343124</GranuleUR>
      <InsertTime>2007-01-05T10:30:30.156Z</InsertTime>
      <LastUpdate>2007-01-05T12:30:30.550Z</LastUpdate>
      <Collection>
        <ShortName>MOD06_L2</ShortName>
        <VersionId>5</VersionId>
      </Collection>
      <Orderable>true</Orderable>
      <AssociatedBrowseImages>
        <ProviderBrowseId>
          BR:036015124
        </ProviderBrowseId>
      </AssociatedBrowseImages>
    </Granule>
  </Granules>
</GranuleMetaDataFile>
```

---

Once ingested, a browse image can be associated with any number of collections or granules. The order of processing followed by ECHO Ingest would allow both the browse image insert and the granule insert above to be placed in the same package and be ingested successfully.

## 3.10 NEW ITEMS VS REPLACE ITEMS

ECHO ingest requires a complete set of collection, granule or browse metadata for both inserts and replaces. When processing collection, granule, or browse metadata, the ECHO system first checks to see if the item already exists. If the item already exists in the system, ECHO will replace the metadata associated with the item with the new metadata. Otherwise, ECHO will consider the ingest to be an insert and load the new item into the ECHO database.

ECHO uses the item identification to search for the existing metadata that needs to be replaced. For a collection, the short name plus version number combined or dataset ID is assumed to be a unique identifier within your metadata. For a granule, the GranuleUR is assumed to be a unique identifier within your metadata. For a browse file, the ProviderBrowseld is assumed to be the unique identifier within your metadata. The ECHO system applies the same principles and validation when dealing with the insertion or replace of the collection, granule, or browse item when processed against XML metadata input.

When replacing collection, granule, or a browse item, only the version with the most recent last update time will be stored in the ECHO database. ECHO will reject any collection, granule, or browse item received that has a last update date pre-dating the records already in the database. If there are duplicated items in one input file the same rule applies, the most recent item will be ingested into ECHO and earlier versions of the item may be reported as rejected, depending on the order of items processed within the file. If the last update time of the item is the same as the one currently in the database, the replacement will be applied. Due to backward compatibility with previous versions of Ingest, the last update time threshold is +/- a half a second.

### Code Listing 39. Metadata Update Example

---

```
<GranulePartialAdd>
  <Targets>
    <Target>
      <GranuleUR>GR:115628</GranuleUR>
    </Target>
  </Targets>
  <Fields>
    <Field>
      <BrowseImage> BR:036015124</BrowseImage>
    </Field>
  </Fields>
</GranulePartialAdd>
```

---

The granule partial add allows a browse image to be associated with an existing granule in ECHO.

## 3.10.1 Inserting Items

### 3.10.1.1 Collections

- a. The default option definition will be set if not specified.
- b. Ingest will generate a unique collection ID with the following format: "C" + unique number + "-" + Data Center ID.
- c. Beginning date/time shall be set to the least of: range date time, single date time, or periodic date time, whichever is specified in the metadata.
- d. Ending date/time shall be set to the greatest of: range date time, single date time, or periodic date time, whichever is specified in the metadata

### 3.10.1.2 Granules

- a. Beginning date/time shall be set to the least of: range date time or single date time, whichever is specified in the metadata.

- b. Ending date/time shall be set to the greatest of: range date time or single date time, whichever is specified in the metadata
- c. Ingest will generate a unique Echo granule ID with the following format: "G" + unique number + "-" + Data Center ID

### **3.10.1.3 Browse**

- a. Browse image files are copied to the public browse path
- b. Browse image filenames must be unique for a provider.

## **3.10.2 Replacing Items**

### **3.10.2.1 Collections**

- a. All data in the collection will be replaced with the metadata of the replacement collection with the exception of
  - 1) ECHO collection ID
  - 2) Granule Spatial Representation
- b. If the provider attempts to replace the Granule Spatial Representation, the collection will be rejected
- c. If a granule references a platform in a collection, then the short names of referenced platforms, instruments, and sensors cannot be removed from the collection and the collection replace will be rejected.
- d. Collection replace will be rejected if the new temporal data invalidates the current beginning/ending data time range of the granules

### **3.10.2.2 Granules**

- a. All data in the granule will be replaced with the metadata of the replacement granule with the exception of
  - 1) ECHO granule ID
- b. Online URL will be replaced

### **3.10.2.3 Browse**

- a. Browse image files are copied to the public browse path
- b. All metadata for the browse image will be replaced with the metadata of the replacement browse image

## **3.11 PARTIALLY UPDATING ITEMS**

If you do not want to replace a complete record in ECHO, ECHO allows you to specify fields to update individually. To partially update a metadata item, the provider has to follow the partial update format included in the ECHO 10 Format. The partial update format requires full metadata only for the particular field that should be updated along with the ID of the item that should be updated. Partial updates are allowed for both granules and collections. Partial updates allow provider to add, update, or delete certain metadata fields such as online access URLs. When processing partial updates, ECHO Ingest applies the same principles and validation as when processing inserts.

Note: When updating Visibility, the update value should be 'true' or 'false' (without the quotes).

### **3.11.1 Collections**

- a. Add/Updateable elements are:
  - 1) Browse references
  - 2) Visibility
  - 3) Temporal
  - 4) Spatial

- 5) Restriction Flag
- 6) Delete Time
- b. Delete-able elements are:
  - 1) Temporal
  - 2) Delete Time
  - 3) Restriction Flag

### 3.11.2 Granules

- a. If a granule has multiple additional attributes with the same name, then all additional attribute values for that name will be replaced with the new values during a partial metadata update.
- b. Add/Updateable elements are:
  - 1) Browse references
  - 2) Online access URL
  - 3) Online resource URL
  - 4) Measured Parameter/QA Flags
  - 5) Additional attribute value
  - 6) Visibility
  - 7) Temporal
  - 8) Spatial
  - 9) Day/Night Flag
  - 10) Cloud Cover
  - 11) Delete Time
  - 12) Restriction Flag
- c. Delete-able elements are:
  - 1) Online access URL
  - 2) Online resource URL
  - 3) Additional attribute value
  - 4) Temporal
  - 5) Day/Night Flag
  - 6) Cloud Cover
  - 7) Measured Parameter/QA Flag
  - 8) Delete Time
  - 9) Restriction Flag

### 3.11.3 Browse

*Partial adds/updates are not supported for browse*

## 3.12 DELETING ITEMS

Instruct ECHO to delete collections, granules, or browse by placing the item ID under the <CollectionDeletes>, <GranuleDeletes>, or <BrowseImageDeletes> tags, respectively. ECHO uses only the item's identification to

---

process the deletion of the item and all the metadata associated with this item. ECHO will keep the deleted items' identification and deletion date in the database for metadata history auditing purposes. The only way to re-install the items in the ECHO system is to re-submit the metadata for those items for insertion.

When a collection is deleted, all associated granules will automatically be deleted. When a browse file is deleted, all collection or granule associations to that browse file will be deleted while the items themselves will remain in ECHO. When collections or granules with browse associations are deleted the browse images will remain in ECHO until a specific browse image delete has been submitted for ingest.

a. Collection deletes cascade to:

- 1) All items owned by the collection (as defined in the schema)
- 2) All granules in the collection

b. Granule

- 1) Deletes cascade to:
  - a) All items owned by the granule (as defined in the schema)
  - b) Browse links/references
- 2) Online URL will be removed

c. Browse deletes cascade to:

- 1) All items owned by the Browse Image (as defined in the schema)
- 2) The browse image file
- 3) Browse links/references

---

#### **Code Listing 40. Collection Deletes**

```
<CollectionMetaDataTable>
    <CollectionDeletes>
        <CollectionDelete>
            <DataSetId>Insert Additional Attributes Not Unique V001</DataSetId>
        </CollectionDelete>
    </CollectionDeletes>
</CollectionMetaDataTable>
```

---

#### **Code Listing 41. Granule Deletes**

```
<GranuleMetaDataTable>
    <GranuleDeletes>
        <DeleteGranule>
            <GranuleUR>SC:MOD021KM.004:19250276</GranuleUR>
        </DeleteGranule>
    </GranuleDeletes>
</GranuleMetaDataTable>
```

---

#### **Code Listing 42. Browse Deletes**

```
<GranuleMetaDataTable>
    <GranuleDeletes>
        <DeleteGranule>
            <GranuleUR>SC:MOD021KM.004:19250276</GranuleUR>
        </DeleteGranule>
    </GranuleDeletes>
</GranuleMetaDataTable>
```

---

```
</GranuleMetaDataFile>
```

### 3.13 ORBIT DATA

Orbit searching is by far the most accurate way to search for level 0-2 orbital swath data. Unfortunately orbital mechanics is a quite difficult field, and the most well known orbit model, the NORAD Propagator, is quite complex. The NORAD Propagator is designed to work with a wide range of possible orbits, from circular to extremely elliptical, and consequently requires quite a bit of information about the orbit to model it well.

To facilitate earth science, the orbits of satellites gathering earth science data are quite restricted compared to the variety of orbits the NORAD Propagator is designed to work with. Generally, the earth science community would like global coverage, with a constant field of view, at the same time every day. For this reason, most earth science satellites are in a sun-synchronous, near-polar orbit. Even missions that are not interested in global coverage, e.g., the Tropical Rainfall Measuring Mission (TRMM), are still interested in having a constant field of view so the coverage of the sensor is at a constant resolution. For this reason, ALL earth science satellites are in circular orbits.

The Backtrack Orbit Search Algorithm, designed and developed by Ross Swick, exploits this fact to simplify the orbit model by modeling an orbit as a great circle under which the Earth rotates. This reduces the number of orbital elements required for the model from 22 to three. Moreover, the NORAD Propagator is designed to predict future orbits based on current status, and consequently must be reinitialized periodically to correct for cumulative error as the model spins forward. As the name implies Backtrack spins the orbit backwards, and in practice spins backwards at most one orbit, so there is no cumulative error.

For more information on Backtrack, please see <http://geospatialmethods.org/bosa/>.

*Note: orbit granules may not be ingested unless the parent collections have orbit parameters defined.*

#### 3.13.1.1 Backtrack Orbit Model

Three parameters to define an orbit:

- Instrument swath width (in kilometers)
- Satellite declination or inclination (in degrees)
- Satellite period (in minutes)

#### 3.13.1.2 Orbit Data Representation

Three parameters to represent orbit data:

- Equatorial crossing longitude
- Start circular latitude (or start latitude and start direction)
- End circular latitude (or end latitude and end direction)

#### 3.13.1.3 How Data Providers Configure Orbit Data

##### Code Listing 43. Add orbit data to Granule Metadata (ECHO format)

```
<Spatial>
  <SpatialDomainContainer>
    <HorizontalSpatialDomainContainer>
      <Orbit>
        <AscendingCrossing>160.14462465545338</AscendingCrossing>
        <StartLat>69.021242</StartLat>
        <StartDirection>D</StartDirection>
        <EndLat>-68.995831</EndLat>
        <EndDirection>A</EndDirection>
```

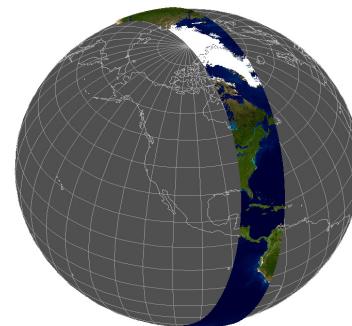


Figure 5. Orbit Swath

```
</Orbit>
</HorizontalSpatialDomainContainer>
</SpatialDomainContainer>

</Spatial>
```

---

#### **Code Listing 44. Add orbit parameters to Collection Metadata (ECHO format)**

---

```
<Spatial>
  <SpatialCoverageType>Horizontal</SpatialCoverageType>
  <OrbitParameters>
    <SwathWidth>400</SwathWidth>
    <Period>98.88</Period>
    <InclinationAngle>98.2</InclinationAngle>
  </OrbitParameters>
  <GranuleSpatialRepresentation>
    <Orbit/>
  </GranuleSpatialRepresentation>
</Spatial>
```

---

## **3.14 INGEST REPORTING**

Ingest generates a report for every job processed, even if the job was deleted. Ingest will automatically copy the report to ECHO's FTP site and will be archived in the job directory for the Ingest Operator. This report will be available for downloading by the Data Provider. ECHO Operations will periodically cleanup the FTP site. While the reports are archived, the archive directory will only be accessible by ECHO Operations. If you need access to a report after it has been removed from the FTP site, you will need to contact ECHO Operations.

The reports are in XML and are designed to be machine-readable and therefore may not be formatted for human consumption. The reports can be used to perform a post processing analysis of a completed job.

### **3.14.1 Ingest Detail Report**

At the completion of each Ingest Job, Ingest will automatically generate an Ingest Detail Report in XML format for the job that details a job's activities as well as any abnormalities discovered in the input file(s). The specific format for the Ingest Detail Report is located on the ECHO website on the Reference Page by selecting "10.0 Ingest Formats" under "Current System" (<http://www.echo.nasa.gov/reference/reference.shtml>).

The Ingest Detail Report will be available in the designated provider report directory (located as agreed upon with ECHO Operations). This directory will be accessible from the ftp site for you to download reports and will be archived in the job directory for the Ingest Operator. (You will not have access to the job directory. If you need access to a report after it has been removed from the ftp site, you will need to contact the Ingest Operator.)

The Ingest Detail Report is applicable to the following types of items:

- Collections
- Granules
- Browse

The Ingest Detail Report consists of two parts: overview and details

### 3.14.1.1 Ingest Detail Report Overview

The overview includes a summary of the items processed:

- Total Processed
- Total Inserted
- Total Replaced
- Total Updated
- Total Deleted
- Total Rejected
- The overview will also include any job-level errors encountered during ingest.

### 3.14.1.2 Ingest Detail Report Details

The following details for each file will be included:

- a. File Name (this is the original file name as sent by the Data Provider)
- b. Processing totals
- c. File errors encountered during ingest.
- d. For each error code (rejection/alert), the following information will be included:
  - 1) Rejection Reason
  - 2) Item ID
    - a) Dataset ID for collections
    - b) Granule UR for granules
    - c) Browse ID for Browse

## 3.15 INGEST ERRORS

There are three different types of errors that may be generated by ECHO Ingest: job errors, file errors, and item errors. Any error that is generated will have a level associated with it: CRITICAL or WARNING. In general, critical errors indicate that the job or part of the job was rejected by ECHO Ingest. Errors that are warnings indicate that some abnormality was detected but the job or affected part of the job may have been ingested dependent on the absence of other critical errors.

### 3.15.1 Job Errors

Initial scanning of a job by ECHO Ingest may reveal problems that will stop processing for that job immediately. If a job with packaged input does not follow the package delivery format or a package's sequence number is out of sync with ECHO Ingest a job error will be generated. Any job error generated by ECHO Ingest is included in the Ingest Detail Report overview. The JobErrorCode type in the Ingest Detail Report Format specifies all possible job errors.

#### Code Listing 45. Job Error Message

---

```
<JobErrors>
  <Error errorCode="DUPLICATE_SEQUENCE_NUMBER" level="CRITICAL">
    <Message>
      The sequence number [300258] is less than or equal to the last
      sequence number seen for the provider.
    </Message>
  </Error>
```

---

```
</JobErrors>
```

### 3.15.2 XML Format Errors

Before ECHO ingests your metadata, it analyzes and validates the metadata against the ECHO Format. If an error is detected, ECHO may create an input file error or an item error. File errors are created when ECHO cannot parse or validate the input file. Item errors are created when ECHO can parse the input file but finds the data invalid. CRITICAL signifies that the entire file or item has been rejected. WARNING signifies that the file or item had some abnormality but may have been ingested depending on the absence of other errors. The ErrorCode type in the Ingest Detail Report Format specifies all possible file errors.

#### 3.15.2.1 FULL\_SCHEMA Error

If an input file does not pass validation against the ECHO Format a FULL\_SCHEMA file error will be generated. To allow the ingest of files that may have some valid and some invalid items, FULL\_SCHEMA errors are considered warnings. An input file with a FULL\_SCHEMA error will be subjected to validation of each individual item against the ECHO Format to determine if any items within the input file are valid.

##### Code Listing 46. FULL\_SCHEMA Error Message

```
<FileErrors>
  <Error errorCode="FULL_SCHEMA" level="WARNING">
    <Message>
      Line:180 Col:140, cvc-minLength-valid: Value '' with length =
      '0' is not facet-valid with respect to minLength '1' for
      type 'GranuleUR'.
    </Message>
  </Error>
</FileErrors>
```

The error message indicates the problem line and column number in the input file. The precise message will vary depending on the type of error found. The sample above indicates that a zero-length GranuleUR was submitted and the ECHO Format specifies that GranuleUR has a minimum length of 1.

#### 3.15.2.2 STRUCTURAL\_SCHEMA Error

If an input file is marked with a FULL\_SCHEMA error ECHO Ingest will attempt to verify that the input file can be parsed by validating the input file conforms the ECHO Format structurally. If an input file is not structurally valid a STRUCTURAL\_SCHEMA error will be generated. STRUCTURAL\_SCHEMA errors are critical. If an input file is structurally valid then each metadata item will be subject to validation against the full ECHO Format, in order to ingest any valid metadata items.

##### Code Listing 47. STRUCTURAL\_SCHEMA Error Message

```
<FileErrors>
  <Error errorCode="FULL_SCHEMA" level="WARNING">
    <Message>
      Line:66 Col:22, cvc-complex-type.2.3: Element
      'GranulePartialAdds' cannot have character [children],
      because the type's content type is element-only.
    </Message>
  </Error>
  <Error errorCode="STRUCTURAL_SCHEMA" level="CRITICAL">
    <Message>
      Line:66 Col:22, cvc-complex-type.2.3: Element
      'GranulePartialAdds' cannot have character [children],
    </Message>
  </Error>
</FileErrors>
```

```
        because the type's content type is element-only.  
    </Message>  
  </Error>  
</FileErrors>
```

---

In the sample above the input was structurally invalid with a character data present in the GranulePartialAdds tag. Such a file cannot be parsed by ECHO Ingest and so a STRUCTURAL\_SCHEMA error is generated and the entire input file is rejected.

### 3.15.2.3 SCHEMA\_VALIDATION\_ERROR Error

If an input file was not valid against the full ECHO Format but was structurally valid then each item will be individually validated against the full ECHO Format. A SCHEMA\_VALIDATION\_ERROR will be generated for any item failing validation and the item will not be ingested. Otherwise, items passing validation will continue normal processing by ECHO Ingest.

#### Code Listing 48. SCHEMA\_VALIDATION\_ERROR Error Message

---

```
<ItemErrorGroups>  
  <ItemErrorGroup errorCode="SCHEMA_VALIDATION_ERROR">  
    <ItemError itemType="COLLECTION"  
      itemId="MLS/Aura L2 Diagnostics, Miscellaneous Grid V001"  
      level="CRITICAL">  
      <Message>  
        Line:14 Col:141, cvc-minLength-valid: Value '' with  
        length = '0' is not facet-valid with respect to  
        minLength '1' for type 'ShortName'.  
      </Message>  
    </ItemError>  
  </ItemErrorGroup>  
</ItemErrorGroups>
```

---

In the sample above the offending collection could be identified and the dataset ID was included in itemId. The collection was not ingested because the ShortName supplied was an empty string and the ECHO Format requires a minimum ShortName length of 1.

### 3.15.2.4 FILE\_TYPE\_INDETERMINABLE Error

If an input file is invalid XML a FILE\_TYPE\_INDETERMINABLE error will be generated. ECHO Ingest will also generate a FILE\_TYPE\_INDETERMINABLE error if an input file is valid XML but ECHO Ingest cannot determine the metadata item type of the file. FILE\_TYPE\_INDETERMINABLE errors are critical.

#### Code Listing 49. FILE\_TYPE\_INDETERMINABLE Error Message

---

```
<FileErrors>  
  <Error errorCode="FILE_TYPE_INDETERMINABLE" level="CRITICAL">  
    <Message>  
      Could not determine file type from input file  
      [EPGMOLT200726120072620101.20070919004641.xml]  
    </Message>  
  </Error>  
</FileErrors>
```

---

### 3.15.3 Data Integrity Errors

All items valid against the ECHO Format is also subject to data integrity validation according to ECHO business rules. For a comprehensive list of the business rules for ingesting data, refer to the ECHO Supplementary Specification by going to the ECHO Development page of the ECHO Website and clicking on "More about Version 10.0": <http://www.echo.eos.nasa.gov/development/index.shtml>.

All metadata items are subject to data integrity validation, including replaces or updates to existing items.

For each data integrity error discovered, ECHO Ingest will include the type of error, the invalid item ID, and the type of item. All data integrity errors are critical errors. The UnionOfItemErrorCode type in the Ingest Detail Report Format specifies all possible data integrity errors.

#### Code Listing 50. Data Integrity Error Message

---

```
<ItemErrorHandler errorCode="TEMPORAL_INVALID_DATE_RANGE">
  <ItemErrorHandler itemType="GRANULE"
    itemId="sample.granule.dat"
    level="CRITICAL">
    <Message>
      The granule's start time [1994-04-09T12:00:00Z] is before
      the collection's [1994-08-16T12:00:00Z]
    </Message>
  </ItemErrorHandler>
</ItemErrorHandlerGroup>
```

---

All data integrity errors will identify the invalid item using itemId and include a detailed message regarding the cause of the error. The precise message will vary depending on the type of error.

## 3.16 NEW ITEMS VS UPDATE ITEMS

ECHO ingest requires a complete set of granule or collection metadata for both inserts and updates. When processing a granule or collection, the ECHO system first checks to see if the item already exists. If the item already exists in the system, ECHO will delete all the metadata associated with this item before inserting the new metadata in its place.

ECHO uses the item identification to search for the existing metadata that needs to be deleted before inserting new metadata. For a granule, the GranuleUR is assumed to be a unique identifier within your metadata. For a collection, the short name plus version number combined is assumed to be a unique identifier within your metadata. The ECHO system applies the same principle when dealing with the insertion or update of the granule or collection when processed against the XML metadata input. When updating a granule or collection, only the version with the most recent update time will be stored in the ECHO database. ECHO will ignore any granule or collection received that has a last update date that pre-dates the records already in the database.

## 3.17 DATA NOT INCLUDED IN THE SCHEMA

Validation not included in the schema is enforced when the data is processed.

For any repetitive item identification in the metadata input, ECHO will ingest into the database only one item bearing the most recent update date with its complete information. The rest of the items and their associated information will be ignored even if they contain different data information and will result in a duplicate error in the ingest summary for that item.

## 3.18 ERROR MESSAGES

During preprocessing, ECHO checks your input files for errors in non-ingest metadata. Then ECHO processes ingest metadata and, upon job completion (successful or not), sends you a summary of the job activity and errors

in the end of job notification email. The Ingest Detail Report details the activities as well as any abnormalities discovered in the input file. The location of the Ingest Detail Report is sent in the end of job notification email.

### 3.18.1 Ingest Detail Report

At the completion of each Ingest Job, Ingest will automatically generate an Ingest Detail Report in XML format for the job with ingest job start/stop time information, data files received, number of inserts/ replaces/ deletions/ updates processed, number of rejections. This report lists all errors encountered while ingesting the metadata itself. See section 3.14.1, Ingest Detail Report, for report details.

### 3.18.2 Input File Validation Error Messages

Before ECHO ingests your metadata, it analyzes and validates them against its corresponding schema. If it detects an error, ECHO rejects the input file and sends an input file error to the email address registered in your ECHO user account. The following is an example of an input file error.

#### Code Listing 51. Input File Validation Error

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="22621990-2B86-9765-67AD-
2A55E4D7EE3B" reportDate="2007-11-30T18:43:20Z">
<Overview endDate="2007-11-30T13:43:16.635-05:00" startDate="2007-11-
30T13:42:44.619-05:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="0" deleted="0" updated="0" 
replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" 
inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" 
inserted="0" processed="0" />
    </ProcessingTotals>
</Overview>
<Details>
    <MetadataFiles>
        <MetadataFile name="TEST_COLL_BROWSE_SINGLE_UPDATE.xml">
            <ProcessingTotals>
                <CollectionProcessingTotals rejected="0" deleted="0" updated="0" 
replaced="0" inserted="0" processed="0" />
                <GranuleProcessingTotals rejected="0" deleted="0" updated="0" 
replaced="0" inserted="0" processed="0" />
                <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" 
replaced="0" inserted="0" processed="0" />
            </ProcessingTotals>
            <FileErrors>
                <Error errorCode="FULL_SCHEMA" level="WARNING">
                    <Message>Line:8 Col:17, cvc-complex-type.2.4.a: Invalid content 
was found starting with element 'LongName'. One of '{":LastUpdate}' is 
expected.
                    </Message>
                </Error>
                <Error errorCode="STRUCTURAL_SCHEMA" level="CRITICAL">
                    <Message>Line:8 Col:17, cvc-complex-type.2.4.a: Invalid content 
was found starting with element 'LongName'. One of '{":LastUpdate}' is 
expected.
                    </Message>
                </Error>
            </FileErrors>
        </MetadataFile>
    </MetadataFiles>
</Details>
```

```
    </MetadataFile>
  </MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.2.1 File\_Type\_Indeterminable

If the input file does not start with the standard XML file declaration line such as <?xml ...>, then this input file is considered an indeterminable file and is rejected for the ingest process. The input file error summary report will list the error with an error message like the one shown below:

**Code Listing 52. Example of File\_Type\_Indeterminable Error**

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="2F8961CA-AA21-B190-63DE-
42C7662902B4" reportDate="2008-02-18T20:32:43Z">
<Overview sequenceNumber="1" endDate="2008-02-18T15:32:42.782-05:00"
startDate="2008-02-18T15:32:33.485-05:00">
  <JobErrors/>
  <ProcessingTotals>
    <CollectionProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
    <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
    <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
  </ProcessingTotals>
</Overview>
<Details>
  <MetadataFiles>
    <MetadataFile name="non-XML.xml">
      <ProcessingTotals>
        <CollectionProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
replaced="0" inserted="0" processed="0" />
      </ProcessingTotals>
      <FileErrors>
        <Error errorCode="FILE_TYPE_INDETERMINABLE" level="CRITICAL">
          <Message>Unable to determine file type of a processing file for
metadata file non-XML.xml
          </Message>
        </Error>
      </FileErrors>
    </MetadataFile>
  </MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.2.2 Invalid XML File

ECHO will validate each input XML file against its corresponding schema. If for any reason the schema validation fails, ECHO will move this file to your backup space for further investigation. The input file error summary report will list the error with an error message like the one shown below:

---

**Code Listing 53. Invalid XML File Error**

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="47C0047E-F506-AD3E-D2F2-
C74F5BB1B45A" reportDate="2008-02-21T20:59:47Z">

<Overview sequenceNumber="10" endDate="2008-02-21T15:59:46.262-05:00"
startDate="2008-02-21T15:59:38.875-05:00">

    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
    </ProcessingTotals>
</Overview>
<Details>
    <MetadataFiles>
        <MetadataFile name="MissingStreetAndCity.xml">
            <ProcessingTotals>
                <CollectionProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
                <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
                <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
replaced="0" inserted="0" processed="0" />
            </ProcessingTotals>
            <FileErrors>
                <Error errorCode="FULL_SCHEMA" level="WARNING">
                    <Message>
                        Line:50 Col:30, cvc-complex-type.2.4.a: Invalid content was
found starting with element 'StateProvince'. One of '{":StreetAddress}' is
expected.
                    </Message>
                </Error>
                <Error errorCode="STRUCTURAL_SCHEMA" level="CRITICAL">
                    <Message>Line:50 Col:30, cvc-complex-type.2.4.a: Invalid content
was found starting with element 'StateProvince'. One of '{":StreetAddress}' is
expected.
                    </Message>
                </Error>
            </FileErrors>
        </MetadataFile>
    </MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.2.3 Missing Manifest File

If a manifest is not included in the package or has an incorrect name (must be "manifest.xml"), the following report is received:

---

**Code Listing 54. Missing Manifest File Error**

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="AETD7" jobId="725D5CD3-B7C7-3F1C-D4D9-9AB2BFA06687"
reportDate="2008-02-20T19:21:58Z">
    <Overview endDate="2008-02-20T14:21:57.995-05:00" startDate="2008-02-
20T14:21:57.398-05:00">
        <JobErrors>

            <Error errorCode="MANIFEST_MISSING" level="CRITICAL">

                <Message>
                    No [manifest.xml] file found in package
                </Message>
            </Error>
        </JobErrors>
        <ProcessingTotals>
            <CollectionProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
            <GranuleProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
            <BrowseProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
        </ProcessingTotals>
    </Overview>
    <Details />
</IngestReport>
```

---

### 3.18.3 Data Ingest Error Messages

When your metadata has been successfully ingested, an Ingest Detail Report is generated detailing the results of the ingest. The location of this report is included in the end of job notification email sent to the email address in your ECHO user account. See section 3.18.1, Ingest Detail Report for more information.

---

**Code Listing 55. Example of the Ingest Summary Report for a Successful Ingest**

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ECHO Ingest Summary Report: Wed Nov 13 13:55:01 EST 2002 --&gt;
<!-- created by ECHO Ingest System --&gt;
&lt;IngestSummary&gt;
&lt;DataProvider&gt;ETE_TEST&lt;/DataProvider&gt;
&lt;IngestStartTime&gt;11/13/2002 13:55:01&lt;/IngestStartTime&gt;
&lt;Collection&gt;
    &lt;InputFiles&gt;
        &lt;File name="VTCCLSR7200211130011240555.XML" size="8424" /&gt;
    &lt;/InputFiles&gt;
    &lt;Processed total="1"&gt;
        &lt;Replacement total="1"&gt;
            &lt;DataSets&gt;
                &lt;DataSet shortname="L7CPF" version="2" /&gt;
            &lt;/DataSets&gt;
        &lt;/Replacement&gt;
    &lt;/Processed&gt;
&lt;/Collection&gt;
&lt;Granule&gt;
    &lt;InputFiles&gt;
        &lt;File name="VTCGLSR7200211130011240101042555.XML" size="1967" /&gt;</pre>

---


```

```
</InputFiles>
<Processed total="2">
    <Insertion total="1" />
    <Replacement total="1" />
</Processed>
</Granule>
<IngestStopTime>11/13/2002 13:56:26</IngestStopTime>
</IngestSummary>
```

---

There are five types of ingest information:

- a. <Browse>
- b. <Valids>
- c. <Collection>
- d. <Granule>
- e. <Updates>

### 3.18.3.1 Out-of-Date Update for Collection or Granule

If the date of a collection or granule in the ECHO database is more recent than the date on the same collection or granule in your input, ECHO will return an error and not process the specific record. In the example below, *n* represents the total number of collections or granules in your input, and *m* represents how many of *n* are out of date (that is, how many pre-date the same collections or granules in the database). If the error occurs at the granule level, the report will reflect only the total number, as represented by *n*.

#### Code Listing 56. Error Returned When Input Data Is Older Than the Information Recorded in the ECHO Database

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="03E8756E-7FA2-D552-EDFD-
BDE96E403213" reportDate="2008-02-21T21:21:58Z">
<Overview sequenceNumber="12" endDate="2008-02-21T16:21:57.665-05:00"
startDate="2008-02-21T16:21:47.936-05:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="1" deleted="0" updated="0"
replaced="0" inserted="0" processed="1" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
    </ProcessingTotals>
</Overview>
<Details>
    <MetadataFiles>
        <MetadataFile name="Collection1ForGranuleInsertOutOfDate.xml">
            <ProcessingTotals>
                <CollectionProcessingTotals rejected="1" deleted="0" updated="0"
replaced="0" inserted="0" processed="1" />
                <GranuleProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
                <BrowseProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
            </ProcessingTotals>
        <FileErrors />
```

```
<ItemErrorGroups>
    <ItemErrorHandler errorCode="OUT_OF_DATE">
        <ItemError itemType="COLLECTION" itemId="Collection1 For Granule
Inserts V001" level="CRITICAL">
            <Message>Validation error, new last update date Tue Oct 10
13:00:00 EDT 2000 is before current last update date Thu Oct 10 13:00:00 EDT
2002
        </Message>
    </ItemError>
</ItemErrorHandler>
</ItemErrorGroups>

</MetadataFile>
</MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.3.2 Invalid Spatial Data

Invalid spatial data will return the sample error shown below. In the example shown below,  $n$  represents the total number of collections or granules in your input, and  $m$  represents how many of  $n$  contain invalid spatial data. If the error occurs at the granule level, the report will reflect only the total number, as represented by  $n$ .

For information on spatial rules for avoiding invalid spatial errors, refer to section 3.5.9, Spatial Representations, Coordinates and Projections and 5.3, Creating Order Options.

---

#### Code Listing 57. Error Returned When the Spatial Coverage Area Data for a Collection or Granule Is Invalid

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="AETD8" jobId="8FB89F41-3352-49E8-9C17-D1B762C53BDD"
reportDate="2008-02-14T21:15:16Z">
<Overview sequenceNumber="40" jobName="40" endDate="2008-02-14T16:15:16.056-
05:00" startDate="2008-02-14T16:15:08.591-05:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="1" deleted="0" updated="0"
replaced="0" inserted="0" processed="1" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0"/>
    </ProcessingTotals>
</Overview>
<Details>
    <MetadataFiles>
        <MetadataFile
name="EDFCMSRT.200804515.200804516.2008045161138.001.001.000040.XML">
            <ProcessingTotals>
                <CollectionProcessingTotals rejected="1" deleted="0" updated="0"
replaced="0" inserted="0" processed="1" />
                <GranuleProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
                <BrowseProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
            </ProcessingTotals>
        </MetadataFile>
    </MetadataFiles>
</Details>
```

```
<FileErrors />
<ItemErrorGroups>
    <ItemErrorHandler errorCode="SPATIAL_INVALID">
        <ItemErrorHandler itemType="COLLECTION" itemId="MISR Level 2 Aerosol
parameters V002" level="CRITICAL">
            <Message>Validation error, Spatial Validation Error
BOUNDING_RECTANGLE #1 [13356 [Element <1>] [Coordinate <1>] [Ring <1>]]
            </Message>
        </ItemErrorHandler>
    </ItemErrorHandlerGroup>
</ItemErrorGroups>

</MetadataFile>

</MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.3.3 Data That Violates Data Constraints

ECHO will reject a granule or collection record with bad data, such as a number attribute that contains a character string. ECHO will return an error like the sample shown below, where  $n$  represents the total number of collections or granules in your input, and  $m$  represents how many of  $n$  contain bad data. If the error occurs at the granule level, the report will reflect only the total number, as represented by  $n$ . Refer to the ECHO data dictionary to avoid bad data errors.

If the data that violates the data constraint appears within the collection's required information, then none of the collection will be ingested into the database. If you notice that the total number of collections processed, as represented by  $n$ , is less than the total number you submitted, contact the ECHO Operations team for investigation.

---

#### Code Listing 58. Error Returned When Any Piece of Data Associated with a Collection or Granule Has Violated a Data Constraint

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="9D3C6D49-FE80-7381-8931-
BC195DA2BCC5" reportDate="2008-02-18T21:10:42Z">
<Overview sequenceNumber="3" endDate="2008-02-18T16:10:41.384-05:00"
startDate="2008-02-18T16:10:34.353-05:00">
    <JobErrors/>
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="1" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0"
inserted="0" processed="0" /></ProcessingTotals>
    </Overview>
<Details>
    <MetadataFiles>
        <MetadataFile name="bad-date.xml">
            <ProcessingTotals>
                <CollectionProcessingTotals rejected="1" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
                <GranuleProcessingTotals rejected="0" deleted="0" updated="0"
replaced="0" inserted="0" processed="0" />
```

```
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        </ProcessingTotals>
        <FileErrors>
            <Error errorCode="FULL_SCHEMA" level="WARNING">
                <Message>Line:6 Col:37, cvc-datatype-valid.1.2.1: 'insert time' is not a valid value for 'dateTime'.
                </Message>
            </Error>
        </FileErrors>
        <ItemErrorGroups>
            <ItemErrorGroup errorCode="SCHEMA_VALIDATION_ERROR">

                <ItemError itemType="COLLECTION" itemId="Test Collection-0" level="CRITICAL">

                    <Message>Line:4 Col:37, cvc-datatype-valid.1.2.1: 'insert time' is not a valid value for 'dateTime'.
                    </Message>
                </ItemError>
            </ItemErrorGroup>
        </ItemErrorGroups>
    </MetadataFile>
</MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.3.4 Invalid Deletion

If your input contains an identifier for metadata that you want to delete, but the identifier does not exist in the database, ECHO will return an error like the sample shown below, where *n* represents the total number of collections or granules in your input, and *m* represents how many of *n* are deletions with non-existent identifiers. If the error occurs at the granule level, the report will reflect only the total number, as represented by *n*.

---

#### Code Listing 59. Error Returned When a Collection or Granule Identifier Sent for Deletion Does Not Exist in the ECHO Database

---

```
<?xml version='1.0' encoding='UTF-8'?><IngestReport dataCenterId="INGEST2" jobId="BF21951A-9AB8-B569-05A9-E1604A4A52F5" reportDate="2008-02-21T22:14:21Z">
<Overview sequenceNumber="16" endDate="2008-02-21T17:14:20.878-05:00" startDate="2008-02-21T17:14:09.697-05:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="1" deleted="0" updated="0" replaced="0" inserted="0" processed="1" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
    </ProcessingTotals>
</Overview>
<Details>
    <MetadataFiles>
        <MetadataFile name="NonExistantGranuleDelete.xml">
            <ProcessingTotals>
```

```
<CollectionProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
    <GranuleProcessingTotals rejected="1" deleted="0" updated="0" replaced="0" inserted="0" processed="1" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
    </ProcessingTotals>
    <FileErrors />
    <ItemErrorGroups>
        <ItemErrorGroup errorCode="GRANULE_NOT_EXISTS">
            <ItemError itemType="GRANULE" itemId="SC:MOD021KM.004:19250276" level="CRITICAL">
                <Message>Validation error, SC:MOD021KM.004:19250276 does not exist.
                </Message>
            </ItemError>
        </ItemErrorGroup>
    </ItemErrorGroups>
</MetadataFile>
</MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.3.5 Granule Associated with a Collection that Does Not Exist in the ECHO Database

An input granule that references a collection that does not exist in the ECHO database will return the sample error shown below, where n represents the total number of collections or granules in your input, and m represents how many of n are granules referencing non-existent collections. This circumstance may occur when the collection being referenced is part of the same round of input but has failed be ingested. If the error occurs at the granule level, the report will reflect only the total number, as represented by n.

---

#### Code Listing 60. Error Returned Because Collection Referenced by Input Granule(s) Does Not Exist in the ECHO Database

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="0BE63912-BF00-86D0-11D7-6FC5C312FDE4" reportDate="2008-02-21T22:22:19Z"><Overview sequenceNumber="17" endDate="2008-02-21T17:22:19.236-05:00" startDate="2008-02-21T17:22:11.231-05:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="1" deleted="0" updated="0" replaced="0" inserted="0" processed="1" />
            <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        </ProcessingTotals>
    </Overview>
    <Details>
        <MetadataFiles>
            <MetadataFile name="GranuleInsertNonExistantCollection.xml">
                <ProcessingTotals>
```

```
<CollectionProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
    <GranuleProcessingTotals rejected="1" deleted="0" updated="0" replaced="0" inserted="0" processed="1" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
    </ProcessingTotals>
    <FileErrors />
    <ItemErrorGroups>
        <ItemErrorGroup errorCode="COLLECTION_REF_INVALID">
            <ItemError itemType="GRANULE" itemId="ScInputGranule001" level="CRITICAL">
                <Message>Referenced collection by short name [Non-Existant-Collection] and version [1] does not exist
                </Message>
            </ItemError>
        </ItemErrorGroup>
    </ItemErrorGroups>
</MetadataFile>
</MetadataFiles>
</Details>
</IngestReport>
```

---

### 3.18.4 Incorrect Date Format

If a provider uses the incorrect data format specified in the provider's schema, the following error occurs:

**Code Listing 61. Incorrect Date Format Error**

---

```
<?xml version='1.0' encoding='UTF-8'?>
<IngestReport dataCenterId="INGEST2" jobId="21A5A8E4-3084-379B-0244-C19DEEDF046E" reportDate="2008-02-21T22:46:27Z">
<Overview sequenceNumber="18" endDate="2008-02-21T17:46:27.099-05:00" startDate="2008-02-21T17:46:18.529-05:00">
    <JobErrors />
    <ProcessingTotals>
        <CollectionProcessingTotals rejected="1" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
    </ProcessingTotals>
</Overview>
<Details>
    <MetadataFiles>
        <MetadataFile name="Collection1ForGranuleInsertBadLastUpdateDateFormat.xml">
            <ProcessingTotals>
                <CollectionProcessingTotals rejected="1" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
                <GranuleProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
            </ProcessingTotals>
        </MetadataFile>
    </MetadataFiles>
</Details>
```

```
        <BrowseProcessingTotals rejected="0" deleted="0" updated="0" replaced="0" inserted="0" processed="0" />
        </ProcessingTotals>
        <FileErrors>
            <Error errorCode="FULL_SCHEMA" level="WARNING">
                <Message>Line:8 Col:52, cvc datatype-valid.1.2.1: '20002-3-15 2:50:00PM' is not a valid value for 'dateTime'.
                </Message>
            </Error>
        </FileErrors>
        <ItemErrorGroups>
            <ItemErrorHandler errorCode="SCHEMA_VALIDATION_ERROR">
                <ItemError itemType="COLLECTION" itemId="Collection1 For Granule Inserts V001" level="CRITICAL">
                    <Message>Line:5 Col:52, cvc datatype-valid.1.2.1: '20002-3-15 2:50:00PM' is not a valid value for 'dateTime'.
                    </Message>
                </ItemError>
            </ItemErrorHandler>
        </ItemErrorGroups>

    </MetadataFile>

</MetadataFiles>
</Details>
</IngestReport>
```

---

# CHAPTER 4. VALIDATING YOUR METADATA

## 4.1 VIEW DATASET INFORMATION

The GetDatasetInformation call gives granule-level details for the requested dataset. The granules must match the given criteria. This method is similar to performing a query; however, it is highly optimized for Data Partner reconciliation of metadata.

Due to the large amount of data that may be returned by this method, FTP Push is the only supported delivery mechanism.

You may specify temporal ranges to limit the search. If specified, the range type field in the granule must be between the start and stop times. If the range type is acquisition, a range intersection (rather than containment) check will be performed. Although the range types could be repeated, there is no benefit to repeating the same range type.

The available online parameters will only return granules if they are available online. If **temporalRanges** is set to false, ECHO will return all matching granules.

All of the restriction fields (dataset ID, ranges, and online flag) will be joined together with the Boolean AND when the search is performed. The standard FTP URL format is:

```
ftp://[user ID:password@]host_name[:port]/[path name/][file name]
```

The file name is ignored and a unique name will be generated by ECHO to ensure uniqueness of the file name.

The following describes the web service call:

```
< GetDatasetInformation>
```

- a. The parameters are defined below:
  - 1) Token: The security token
  - 2) dataSetId: The ID of the dataset from where the information comes
  - 3) temporalRanges: The temporal ranges to apply when searching
  - 4) availableOnline: This should be set to true to restrict results to granules available online, or to false to get all granules regardless of online status.
  - 5) browseAvailable: This should be set to true to restrict the results to granules which have browse data available online, or false to get all granules regardless of browse data availability.
  - 6) ftpUrl: The FTP location to push the results to
- b. Returns:
  - 1) The unique filename that will be created on the FTP server as the results are processed

**Code Listing 62. View Dataset Information**

```
<s0:GetDatasetInformation
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:echo="http://echo.nasa.gov/echo/v9/types"
xmlns:s0="http://echo.nasa.gov/echo/v9">

<s0:token>TOKEN_ABCDEFG12345</s0:token>

<s0:datasetId>MODIS_EXAMPLE_V001</s0:datasetId>
```

```
<s0:temporalRanges>

  <echo:Item>

    <echo:StartDate>2006-04-13T23:45:00.000Z</echo:StartDate>

    <echo:StopDate>2007-04-13T16:19:04.921Z</echo:StopDate>

    <echo:TargetTemporalField>PRODUCTION</echo:TargetTemporalField>

  </echo:Item>

</s0:temporalRanges>

<s0:availableOnline>true</s0:availableOnline>

<s0:browseAvailable>true</s0:browseAvailable>

<s0:ftpUrl>ftp://user:mypassword@myserver:21/echo/result</s0:ftpUrl>

</s0:GetDatasetInformation>
```

---

# CHAPTER 5. CONTROLLING ACCESS TO YOUR METADATA

## 5.1 MANAGING DATA ACCESS RULES

Use the Provider User Management Program (PUMP) or the API itself to set up Data Access Rules as discussed in this chapter. To access PUMP, go to <https://api.echo.nasa.gov/pump> for the Operational system and <https://api-test.echo.nasa.gov/pump> for the partner test system. See the *ECHO 10 PUMP User Guide* for information on how to use PUMP.

You can manage access to your metadata using the Data Management Service by creating **DataAccessRules** that restrict or permit ECHO users to take actions on the data. Supported **Actions** include Viewing, Browsing, and Ordering metadata.

In addition to the restrictions and permissions, a collection can be set to be restricted from viewing, browsing, and ordering by all users except the ones acting on behalf of the provider who owns the collection by simply setting its visibility flag . This restriction can be removed by simply changing its visibility flag back to open to the public. The restriction applied on the collection will also be applied on all its granules.

## 5.2 DATA ACCESS RULES

**DataAccessRules** can be applied to specific groups of ECHO users to give data access permission to users who have similar access privileges. You may use groups to establish permission data access rules. If you have not specified any groups in **DataAccessRules**, then the rules apply to every ECHO user.

A single **DataAccessRule** applies to collections or granules, not both. A set of **DataValue** values in a **DataAccessRule** specifies which collections or granules the rule covers. If the rule is for collections, then **DataValue** should contain dataset IDs. If the rule is for granules, then **DataValue** should contain granule URs (Universal Reference). If you have not specified any value, then the rule applies to both collections and granules.

Conditions can further restrict which collections or granules the data rule covers. There are four types of conditions:

- a. Temporal
- b. Rolling Temporal
- c. Restriction Flag
- d. Boolean

Use the **ConditionComparator** in the data access rule to compare the condition values with the metadata value.

**Temporal** and **Rolling Temporal** conditions are based on time fields in the metadata. **Temporal** specifies a start and end time. The fields **Production date**, **Acquisition date**, **Insert date**, and **Last Update date** can be used for Temporal conditions but only with the equals and not equals comparators.

**Rolling Temporal** specifies an amount of time after the current time and uses the same fields as Temporal. Rolling Temporal can only be used with the greater than, greater than or equal to, less than, and less than or equal to comparators.

**Restriction Flag** indicates that the data access rule should be based on the value of the restriction flag in the metadata. The field “Lower” in the Restriction Flag condition indicates a value for the restriction flag field in the metadata. Restriction Flag conditions can be used with any comparator.

**Boolean** is a very basic condition that you should use if you do not want one of the other conditions. It has a single Boolean flag, and if the flag is true, then the data access rule will be applied. If the flag is false, then the data access rule will not be applied. Setting the flag to false is equivalent to not creating a data access rule. Boolean conditions can only be used with the equals comparator.

### Code Listing 63. Data Access Rules Examples

```
<BooleanConditionField>
```

```
<ConditionName>True condition</ConditionName>
<Description>This is a true data condition. False condition use BooleanFlag
set to be false.</Description>
<BooleanFlag>true</BooleanFlag>
</BooleanConditionField>

<TemporalConditionField>
<ConditionName>year 2006</ConditionName>
<Description>This is year 2006 data. Use StartTime and StopTime for the
correct range. Use TargetTemporalField to specify temporal type. Possible enums
are PRODUCTION, ACQUISITION, INSERT, LAST_UPDATE.</Description>
<StartTime>2006-01-01T00:00:00.00Z</StartTime>
<StopTime>2006-12-31T11:59:59.00Z</StopTime>
<TargetTemporalField><PRODUCTION/></TargetTemporalField>
</TemporalConditionField>

<RollingTemporalConditionField>
<ConditionName>30 days duration</ConditionName>
<Description>This is 30 days duration condition. Use Duration to specify
duration in milliseconds: 30x24x60*60*100 = 259200000.</Description>
<Duration>259200000</Duration>
<TargetTemporalField><PRODUCTION/></TargetTemporalField>
</RollingTemporalConditionField>

<RestrictionFlagConditionField>
<ConditionName>Restriction flag lower than 3</ConditionName>
<Description>This condition is used to restrict access to data whose
restriction flag is lower than 3</Description>
<Lower>3</Lower>
</RestrictionFlagConditionField>
```

---

## 5.3 WIST VALIDS

A full set of WIST valids is run automatically at least once per day. The WIST valids are filtered based on what collections the user has permission to see. For guest users the collection filter is updated every 30 minutes - so if a collection becomes visible (or restricted) to a guest user - that will be reflected in WIST within 30 minutes. For registered users, their collection filter is created when the user logs in to WIST - any collections visible to that user will be in his collection filter. If a collection becomes visible to a user after they have logged into WIST - then it will not be visible to him until the next time he logs into WIST.

WIST url:

<http://wist.echo.nasa.gov/~wist/api/mswelcome/>

# CHAPTER 6. CREATING ORDER OPTIONS

Order options allow you to configure a block of custom provider policies that will define the information that a client must provide as part of an order, as well as other settings such as communications and firewall rules. Refer to the next chapter for details on communication configuration.

Data Partners generally use Provider User Management Program (PUMP) or the API itself to set up order options. To access PUMP, go to <https://api.echo.nasa.gov/pump> for the Operational system and <https://api-test.echo.nasa.gov/pump> for the partner test system. See the *ECHO 10 PUMP User Guide* for information on how to use PUMP.

## 6.1 ORDER OPTIONS

The option hierarchy in ECHO was created to be as flexible as possible in terms of Data Partner ordering needs. For details about ECHO Order Fulfillment, refer to the Data Partner Tools Page of the ECHO website at: [http://www.echo.nasa.gov/data\\_partners/data\\_tools.shtml](http://www.echo.nasa.gov/data_partners/data_tools.shtml).

*Note: Not all catalog items have associated definitions – in which case they do not need option selections when ordering. However, if there are any options that are required by that catalog item, they will have to be filled out before you can validate and/or submit the order (or else an error will be returned).*

Although you can create order options in PUMP, the output will be in ECHO Forms. For details about ECHO Forms and the ECHO Forms schema, refer to the Data Partner Tools Page of the ECHO website at: [http://www.echo.nasa.gov/data\\_partners/data\\_tools.shtml](http://www.echo.nasa.gov/data_partners/data_tools.shtml).

### 6.1.1 Assigning Option Definitions to Catalog Options

#### 6.1.1.1 OptionDefinitions

Option Definitions allow third parties to define parts of the API on ECHO using XML. Option Selections contain XML that must conform to the Option Definitions.

#### 6.1.2 Elements

Option assignment calls take a list of option definitions described below. Refer to this list of the major elements used in Option Definitions, which shows either a short definition of the element or the web page that defines them in detail. For more information on types and elements, see the ECHO 10 Web Services API at <http://api.echo.nasa.gov/echo/ws/v10/index.html> for the operational system and <http://testbed.echo.nasa.gov/echo/ws/v10/index.html> for testbed..

- a. **Guid:** This is a unique ID assigned by ECHO for the definition (String, Multiplicity: 0 to 1, unique in ECHO)
- b. **Name:** This is the name of the definition (String, Multiplicity: 1 to 1, unique per provider)
- c. **Scope:** The scope of the definition; used to determine if the option is a system-level option or a provider level option. Only administrators may add system-level definitions and only Data Providers may add provider-level definitions. (Enumarated, Multiplicity: 1 to 1)
  - 1) **OptionScope:** This defines the different scopes for an option definition
- d. **Deprecated:** Indicates if the definition is deprecated (obsolete). Deprecated definitions will be returned to the client and will be considered valid when validating an order; however, new order items cannot be added using the definition and existing order items cannot be updated using the definition. This flag is ignored when creating an option definition; however, it will always be set and returned once the definition has been created in ECHO. (Multiplicity: 0 to 1)

- e. **Form:** The ECHO Form itself. Contents must conform to ECHO Forms schema. See the ECHO Forms Specification for more information on the Data Partner Tools page of the ECHO website ([http://www.echo.nasa.gov/data\\_partners/data\\_tools.shtml](http://www.echo.nasa.gov/data_partners/data_tools.shtml)). (Multiplicity: 1 to 1)

- 1) **AnyElement:** Any single XML element.

### 6.1.3 Calls

These operations require Data Provider or Administrator access. They add the given option definitions to the pool of order options for provider. Use order options to collect provider-specific information for orders. For more information about calls on the ECHO 10.0 Web Services API page at <http://api.echo.nasa.gov/echo/ws/v10/index.html> for the operational system and <http://testbed.echo.nasa.gov/echo/ws/v10/index.html> for testbed.

a. Faults

- 1) **Argument** – Used to specify ArgumentName and its value that caused a fault.
- 2) **AuthorizationFault** – Reported by ECHO when a user is not authorized to invoke an operation
- 3) **DataSizeLimitFault** – Reported by ECHO to indicate that a data size limit has been exceeded
- 4) **DuplicateIdFault** – Reported by ECHO to indicate that an entity with the same ID already exists in ECHO
- 5) **ECHOFault** – Provides information about the error that occurred during the invocation of an ECHO operation
- 6) **InternalFault** – Reported by ECHO when an internal error occurs
- 7) **InvalidArgumentFault** – Reported to indicate that one or more arguments passed were invalid
- 8) **InvalidStateFault** – Reported to indicate that an action by the client would put an object in ECHO in an invalid state
- 9) **InvalidURLFault** – Reported to indicate invalid syntax in a URL or an element of the URL that does not exist
- 10) **ItemNotFoundFault** – Reported when the client attempts to access one or more objects that do not exist
- 11) **ListOfArguments** – Used to specify a list of arguments and their values that caused a fault
- 12) **ParseFault** – Reported to indicate that some value could not be parsed
- 13) **RemovalFault** - Reported to indicate an error occurred during the removal of an object from ECHO
- 14) **UnsupportedFeatureFault** - Reported to indicated that a feature was selected that is not supported
- 15) **ValidationFault** - Reported to indicate that an object in or passed to ECHO is not valid

b. Parameters

- 1) **token:** The Security Token with Provider Role or administrator
- 2) **optionDefinitions:** The option definitions to be added—at least one is required.

c. Returns

- 1) ECHO returns a list of GUIDs identifying the created option definitions.

The GUID assigned is a unique filename that will be created on the FTP server as the results are processed.

# CHAPTER 7. SETTING UP COMMUNICATION CONFIGURATIONS FOR ORDERS

## 7.1 GENERAL INFORMATION FOR CONFIGURATION

Most Data Partners use the web-based Provider User Management Program (PUMP) to configure communications for orders (that is, to be able to capture orders through ECHO). To access PUMP, go to <https://api.echo.nasa.gov/pump> for the Operational system and <https://api-test.echo.nasa.gov/pump> for the partner test system. See the *ECHO 10 PUMP User Guide* for information on how to use PUMP.

## 7.2 WHAT MUST BE SPECIFIED

### 7.2.1 Provider Policies

Establish a set of Data Provider policies for orders using PUMP. Each provider has a set of provider policies that contain information such as the endpoint for the provider (or adapter), the number of attempts to complete the transaction, which operations are supported, and the amount of time between attempts. These definitions are standardized to easy-to-understand business objects.

#### 7.2.1.1 Endpoint for Receiving Orders

You must specify an “end point,” a Uniform Resource Identifier (URI), also known as network address, where ECHO can send the order. The network address is usually either an IP or HTTP address. Within PUMP, select Provider Policies and type the URI into the End Point text box (under routing).

You need to make the exact URI location available for each of the user actions that you support. This location is where ECHO ultimately sends the actual transaction message.

#### 7.2.1.2 Secure Socket Layer (SSL)

The Order Fulfillment API fully supports SSL encrypted order transmissions because ECHO supports certificate authority (CA) signed certificates, such as Verisign certificates, as well as self-signed certificates.

Set the certificate information. The basic steps in configuration of an SSL are:

- d. The certificate must be generated or purchased.
- e. The server hosting your endpoint must be configured to use SSL.
- f. The provider policies must be configured with the public, PEM-encoded key from the certificate.

For more information and Order Options Samples (XML), refer to the Data Partner Tools page on the ECHO website: [http://www.echo.nasa.gov/data\\_partners/data\\_tools.shtml](http://www.echo.nasa.gov/data_partners/data_tools.shtml).

## 7.3 RECEIVING AND FULFILLING ORDERS

The following information refers to standard, relevant fields in most Data Partner’s Order Options that you, as a Data Partner, will read when you receive an order. This information is generic. Most Data Partners would consider these standard options; however, your operation may use different options. For that reason, the examples shown after the descriptions of each standard field do not necessarily correspond to your own configuration.

### 7.3.1 Order Fulfillment

For Order Fulfillment API Documentation, Order Fulfillment Types XML Schema, and the Order Fulfillment Service WSDL, refer to the Data Partner Tools page on the ECHO website:  
[http://www.echo.eos.nasa.gov/data\\_partners/data\\_tools.shtml](http://www.echo.eos.nasa.gov/data_partners/data_tools.shtml).

### 7.3.2 Order Identifier

The OrderId provides identifying information about an order.

- The **ProviderId** is ECHO's name for you, the Data Partner.
- The **OrderId** is ECHO's unique identifier (a GUID) for the order.
- The **TrackingId** is where your unique order tracking number is maintained for future reference.
- The **DataCenterId** is the name by which the provider expects to be called when receiving an order.

```
<complexType name="OrderId">
  <sequence>
    <element name="ProviderId" type="string"/>
    <element name="OrderId" type="string"/>
    <element name="TrackingId" type="string"/>
    <element name="DataCenterId" type="string"/>
  </sequence>
</complexType>
```

### 7.3.3 Line Items

The Line Item represents the item to be ordered (collection or granule) and all supporting information describing exactly what is being ordered.

- The first field of a Line Item is the **GranuleUR** (granule universal reference), which is your unique identifier for the granule. This is your unique identifier, not ECHO's unique identifier. This is optional if the Line Item being ordered is a data set.
- The next field is the **DataSetId**. This is the provider's unique identifier for a collection. It will be filled in with the identifier of the collection to which the granule belongs if the order is for a granule, and the identifier of the collection being ordered if the order is for a collection.
- The **Quantity** field allows an order to specify how many of an item should be ordered. If the order is for electronic delivery (that is, FTP), then the quantity should always be one. If the order is for the delivery of pre-packaged media, then the quantity can be considered the number of media that are being requested.
- The **AuthenticationKey** field is set based on the client calling SetAuthenticationKey. (This field can be no longer than 16 characters.)
- Finally, the options specified by the user regarding this line item are passed in as **OrderOptions**.

```
<complexType name="ListOfLineItems">
  <sequence>
    <element name="lineItems" type="tns:LineItem" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<complexType name="LineItem">
  <sequence>
    <element name="GranuleUr" type="string" minOccurs="0"/>
    <element name="DataSetId" type="string"/>
    <element name="Quantity" type="integer"/>
    <element name="AuthenticationKey" type="string" minOccurs="0"/>
    <element name="OrderOptions" type="options:OptionSelection"
      minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

### 7.3.4 User Information

User Information conveys who is placing the order. It consists of the ECHO User ID and the associated addresses for the order. While only the Contact Address is required, it should be noted that some Data Providers require all three addresses to be completed in order to fulfill the order.

```
<complexType name="UserInformation">
  <sequence>
    <element name="EchoUserId" type="string"/>
    <element name="ShippingAddress"
      type="tns:ShippingAddress" minOccurs="0"/>
    <element name="BillingAddress"
      type="tns:BillingAddress" minOccurs="0"/>
    <element name="ContactAddress"
      type="tns:ContactAddress"/>
  </sequence>
</complexType>
```

#### 7.3.4.1 Shipping Address

Use the Shipping Address to indicate where an order should be physically shipped. This is not applicable if you are using FTP for order fulfillment.

- The **Name field** contains the first and last name of the person to whom the order should be shipped.
- The **Business Name** contains the name of the company to whom the order should be shipped; this is an optional field.
- The **Contact Name** is the name of the person to be contacted if there are any questions about shipping.
- The **Phone Number** contains a string with the telephone number of the contact name in case there are shipping issues.
- The **Special Instruction** field allows for more information for the delivery (for example, “Leave the package in front of the door.”)
- Finally, the **Address** field is the destination of the delivery.

```
<complexType name="ShippingAddress">
  <sequence>
    <element name="Name" type="tns:Name"/>
    <element name="BusinessName" type="string" minOccurs="0"/>
    <element name="ContactName" type="string"/>
    <element name="PhoneNumber" type="string"/>
    <element name="SpecialInstruction" type="string"/>
    <element name="EmailAddress" type="string"/>
    <element name="Address" type="tns:Address"/>
  </sequence>
</complexType>
```

#### 7.3.4.2 Billing Address

The Billing Address is similar to the Shipping Address, and identifies who will be paying the bill for the order. There is no Special Instruction field for this case.

```
<complexType name="BillingAddress">
  <sequence>
```

```
<element name="Name" type="tns:Name"/>
<element name="BusinessName" type="string" minOccurs="0"/>

<element name="ContactName" type="string"/>
<element name="PhoneNumber" type="string"/>
<element name="EmailAddress" type="string"/>
<element name="Address" type="tns:Address"/>
</sequence>
</complexType>
```

---

#### 7.3.4.3 Contact Address

The Contact Address is also similar to the Shipping Address, but it identifies who is placing the order. Again, there is no Special Instruction field.

---

```
<complexType name="ContactAddress">
<sequence>
<element name="Name" type="tns:Name"/>
<element name="PhoneNumber" type="string"/>
<element name="EmailAddress" type="string"/>
<element name="BusinessName" type="string"/>
<element name="Address" type="tns:Address"/>
</sequence>
</complexType>
```

---

#### 7.3.4.4 Name

This structure is simply a way to distinguish the first and last name of a person. You can append a middle initial or middle name to First Name if needed. You can also add a suffix to Last Name, such as Jr.

---

```
<complexType name="Name">
<sequence>
<element name="FirstName" type="string"/>
<element name="LastName" type="string"/>
</sequence>
</complexType>
```

---

#### 7.3.4.5 Address

The address field is a structure that describes an address.

- The **ID** field is a name that the user uses to describe the address (for example: Home, Work, Project1, etc.)
  - The **US Format** flag is used to indicate whether a provider can rely on the address looking like a standard US address
    - The rule is that the system will validate those addresses that have the US Format flag set by checking that Street1, City, State, Zip Code, and Country are all set
    - If the US Format flag is clear, then only Street1 and Country are required.
    - ECHO allows for up to 5 street address lines in either case.
- 

```
<complexType name="Address">
<sequence>
```

---

```
<element name="Id" type="string"/>
<element name="USFormat" type="boolean"/>

<element name="Street1" type="string"/>
<element name="Street2" type="string" minOccurs="0"/>
<element name="Street3" type="string" minOccurs="0"/>
<element name="Street4" type="string" minOccurs="0"/>
<element name="Street5" type="string" minOccurs="0"/>
<element name="City" type="string" minOccurs="0"/>
<element name="State" type="political:State" minOccurs="0"/>
<element name="ZipCode" type="string" minOccurs="0"/>
<element name="Country" type="political:Country"/>
</sequence>
</complexType>
```

#### 7.3.4.6 Client Identity

The client identity is unique for each ECHO client. Orders coming from an ECHO Client Partner carry this unique ID. Use the client identity to determine the source of any given order. This information is sent along with the other order information to the provider via the provider proxy. The client identity takes the form of a string generated by the client and is expected to look like "Client Name v1.0."

*Note: Client identities are less than 16 characters.*

## APPENDIX A. ACRONYMS USED IN ECHO

Acronyms used in this document are contained in this appendix. A complete list of the acronyms frequently used in discussions of ECHO can be found on the ECHO website on the ECHO Acronyms page at [http://www.echo.nasa.gov/overview/over\\_acronyms.shtml](http://www.echo.nasa.gov/overview/over_acronyms.shtml).

ACL	Access Control List
API	Application Programming Interface
AQL	Alternative Query Language
ASF DAAC	Alaska Satellite Facility DAAC
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
BMGT	Bulk Metadata Generation Tool
COTS	Commercial Off The Shelf
DAAC	Distributed Active Archive Center
DB	DataBase
DTD	Document Type Definition
ECHO	EOS Clearinghouse
ECS	EOSDIS Core System
EDC	EROS Data Center
EDG	EOS Data Gateway
EJB	Enterprise JAVA Beans
EMD	EOSDIS Maintenance and Development
EOS	Earth Observing System
EOSDIS	EOS Data and Information System
EROS	Earth Resources Observation Systems
ESDIS	Earth Science Data and Information System
ESIP	Earth Science Information Partner
ETC	ECHO Technical Committee
FTP	File Transfer Protocol
GCMD	Global Change Master Directory
GES DAAC	GSFC Earth Sciences DAAC
GHRC	Global Hydrology Resource Center
GIS	Geographic Information System
GML	Geography Markup Language
GMT	Greenwich Mean Time
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
GUID	Globally Unique Identifier

---

IIMS	Independent Information Management Subsystem
J2EE	Java 2 Enterprise Edition
LAADS	Level 1 and Atmosphere Archive and Distribution System
LP DAAC	Land Processes DAAC
MISR	Multiangle Imaging SpectroRadiometer
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NSIDC DAAC	National Snow and Ice Data Center DAAC
ODL	Object Description Language
OGC	OpenGIS Consortium
ORNL DAAC	Oak Ridge National Laboratory DAAC
PGE	Product Generation Executives
PO.DAAC	Physical Oceanography DAAC
PSA	Product Specific Attribute
PUMP	Provider User Management Program
QA	Quality Assurance
SEDAC	Socioeconomic Data and Applications Center
SOAP	Simple Object Access Protocol
SSC	Stennis Space Center
SSL	Secure Sockets Layer
UDDI	Universal Description, Discovery, and Integration
UI	User Interface
UR	Universal Reference
UR	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Universal Time, Coordinated (also called GMT/UTC)
WIST	Warehouse Inventory Search Tool
WGS	World Geodetic System
WRS	Worldwide Reference System
WSDL	Web Services Description Language
XML	eXtensible Markup Language
XSLT	eXtensible Style Language Transformation

**APPENDIX B. FUNCTIONAL BREAKDOWN BY USER/ROLE TYPE**

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
<b>AdministratorService</b>				
GetAvailableMetricReports				
GetErrorMessages	√	√	√	√
GetMetricReports				
GetSecurityTokenInformation				
InitializeEchoSystemuser				
InitializeErrorCodes				
InitializeTaxonomies				
PurgeArchiveRecords				
RemoveMetricReports				
RemoveTokens				
RevokeAllTokensForUsers				
RevokeTokens				
SetErrorMessages				
SysncUddiRegistry				
<b>AuthenticationService</b>				
GetECHOVersion	√	√	√	√
GetSecurityTokenInformation				
login		√	√	√
logout		√	√	√
RemoveTokens				
RevokeAllTokensForUsers				
ReokeTokens				
<b>CatalogService</b>				
CancelAsynchronousQuery		√	√	√
ExecuteQuery	√	√	√	√
ExecuteSavedQuery		√	√	√

Functional Breakdown By User/Role Type

<b>Service/Transaction</b>	<b>Guest User</b>	<b>Registered User</b>	<b>Registered User with Data Provider Role</b>	<b>Registered User with Client Provider Role</b>
GetCatalogItemMetadata	✓	✓	✓	✓
GetQueryResults	✓	✓	✓	✓
GetSavedQueries		✓	✓	✓
GetSavedQueryNames		✓	✓	✓
GetSavedSavedResultSetNames		✓	✓	✓
RemoveSavedQueries		✓	✓	✓
RemoveSavedResultSets		✓	✓	✓
ResolveMetadataPaths			✓	
SavedResultSet		✓	✓	✓
SaveQuery		✓	✓	✓
<b>DataManagementService</b>				
All			✓	
<b>EventNotificationService</b>				
All		✓	✓	✓
<b>ExtendedServicesService</b>				
ActivateWSAdvertisements				
ActivateWSGuis				
ActivateWSImplementations				
ActivateWSInterfaces				
CreateWSAdvertisements				✓
CreateWSGuis				✓
CreateWSImplementations				✓
CreateWSInterfaces				✓
GetOperationsByImplementation	✓	✓	✓	✓
GetServiceNamesByTaxonomyEntry	✓	✓	✓	✓
GetWSAdvertisements	✓	✓	✓	✓
GetWSDLForWSInterface	✓	✓	✓	✓
GetWSGuis	✓	✓	✓	✓
GetWSImplementationNamesByWSInterface	✓	✓	✓	✓

## Functional Breakdown By User/Role Type

<b>Service/Transaction</b>	<b>Guest User</b>	<b>Registered User</b>	<b>Registered User with Data Provider Role</b>	<b>Registered User with Client Provider Role</b>
GetWSImplementations	✓	✓	✓	✓
GetWSInterfaces	✓	✓	✓	✓
RemoveWSAdvertisements				✓
RemoveWSGuis				✓
RemoveWSImplementations				✓
RemoveWSInterfaces				
UpdateWSAdvertisements				✓
UpdateWSGuis				✓
UpdateWSImplementations				✓
UpdateWSInterfaces				
<b><i>GroupManagementService</i></b>				
CreateGroups			✓	
GetGroupNames		✓	✓	✓
GetGroupNamesByManager		✓	✓	✓
GetGroupNamesByName		✓	✓	✓
GetGroups		✓	✓	✓
NotifyManagers		✓	✓	✓
NotifyMembers		✓	✓	✓
RemoveGroups (only group manager)		✓	✓	✓
UpdateGroups (only group manager)		✓	✓	✓
<b><i>OrderManagementService</i></b>				
AddOrderItems	✓	✓	✓	✓
CancelOrder	✓	✓	✓	✓
CancelProviderOrder	✓	✓	✓	✓
CreateAndSubmitOrder	✓	✓	✓	✓
CreateOrder	✓	✓	✓	✓
GetOrderItemNamesByOrder	✓	✓	✓	✓
GetOrderItemNamesByProviderOrder	✓	✓	✓	✓
GetOrderItems	✓	✓	✓	✓
GetOrders	✓	✓	✓	✓

Functional Breakdown By User/Role Type

<b>Service/Transaction</b>	<b>Guest User</b>	<b>Registered User</b>	<b>Registered User with Data Provider Role</b>	<b>Registered User with Client Provider Role</b>
GetProviderOrderGuidsByStateAndOwner		√	√	√
GetProviderOrderGuidsByStateAndProvider			√	√
GetProviderOrderGuidsByStateDateAndProvider			√	√
QuoteOrder	√	√	√	√
RemoveOrderItems	√	√	√	√
RemoveOrders	√	√	√	√
RemoveProviderOrders	√	√	√	√
SetAuthenticationKey		√	√	√
SetUserInformationForOrder	√	√	√	√
SubmitOrder	√	√	√	√
UpdateOrderItems	√	√	√	√
ValidateOrder	√	√	√	√
<b><i>OrderProcessingService</i></b>				
All			√	
<b><i>ProviderService</i></b>				
ActivateProvider				
AddAuthenticatorDefinitions			√	
CreateProvider		√	√	√
GetAuthenticatorDefinitions	√	√	√	√
GetProviderNames	√	√	√	√
GetProviderPolicies			√	√
GetProviders	√	√	√	√
GetProviderSupportedTransactions	√	√	√	√
GetProviderNamesByProviderId	√	√	√	√
RemoveAuthenticatorDefinitions			√	
RemoveProviderPolicies			√	
SetProviderPolicies			√	
UpdateProvider			√	√

Functional Breakdown By User/Role Type

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
<b>SubscriptionService</b>				
All		√	√	√
<b>TaxonomyService</b>				
AddTaxonomyEntry				
CreateTaxonomy				
CreateVirtualTaxonomy				
GetRootPath	√	√	√	√
GetTaxonomies	√	√	√	√
GetTaxonomyEntries	√	√	√	√
GetTaxonomyEntry	√	√	√	√
<b>UserService</b>				
CreateAuthenticator		√	√	√
CreateUser	√			
GetAuthenticators		√	√	√
GetAuthenticatorNames		√	√	√
GetCurrentUser		√	√	√
GetUserNames		√	√	√
GetUserNamesByUserId		√	√	√
GetUserNamesByOrganizationAndFirstAndLastName			√	√
GetUserNamesByRole (provider role)			√	√
GetUserPreferences		√	√	√
GrantAccess (provider access)			√	
RecallUserId	√	√	√	√
RemoveAuthenticators		√	√	√
RemoveUserPreferences		√	√	√
RemoveUsers				
ResetUserPassword	√	√	√	√
RevokeAccess (provider access)			√	
SetUserPassword		√	√	√

Service/Transaction	Guest User	Registered User	Registered User with Data Provider Role	Registered User with Client Provider Role
SetUserPreferences		√	√	√
UpdateUser		√	√	√

## APPENDIX C. PRE-ECHO 10 INGEST REFERENCE

ECHO Ingest will continue to support metadata input from existing Data Partners that have not transitioned to the ECHO 10 Format. While data conforming to the ECHO 10 Format does not vary significantly from previous versions, the following information serves as reference material for Ingest aspects that do not apply to the ECHO 10 Format.

All Data Partners, regardless of the metadata input format, will have their metadata subject to the same data constraints and data integrity validation as detailed throughout Chapter 3, Ingest. In addition, all Data Partners will receive the same Ingest Detail Report described in section 3.18.1 Ingest Detail Report

### C.1 METADATA INGEST DTD

If you do not want to replace the complete record in ECHO with an update, ECHO allows you to specify fields to update individually. To update metadata, the provider has to follow ECHO's metadata update DTD and some other rules for the metadata update XML files.

Currently, ECHO handles granule-level Online Access URL (delete, update, insert) and Measured Parameter QA Flags (update) metadata update and browse update. ECHO ignores collection-level metadata updates.

#### Code Listing 64. ECHO 8.0 Metadata Update DTD

---

```
?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Keith Wichmann
(Global Science and Technology, Inc.) --&gt;
&lt;!ELEMENT ProviderAccountService (UpdateMetadata)&gt;

&lt;!ATTLIST ProviderAccountService xmlns CDATA #IMPLIED&gt;

&lt;!--UpdateMetadata can update a single collection, multiple collections, a
single granule, or multiple granules in one transaction. Each update allows the
addition of new metadata--&gt;
&lt;!ELEMENT UpdateMetadata (Collection*, Granule*)&gt;
&lt;!ELEMENT Collection (Target+, (Add | Update | Delete)+)&gt;
&lt;!ELEMENT Granule (Target+, (Add | Update | Delete)+)&gt;
&lt;!-- Target+ allows the same change to be made to several different granules or
collections simultaneously. This is especially useful for bulk deletions of
OnlineURLs. --&gt;
&lt;!ELEMENT Target (ID, ProviderLastUpdateDateTime, SaveDateTimeFlag?)&gt;
&lt;!-- SaveDateTimeFlag is the flag that allows echo to update the last update
date time for Target. The default is SAVE --&gt;
&lt;!ELEMENT Add (QualifiedTag, MetadataValue)&gt;
&lt;!ELEMENT Update (QualifiedTag, MetadataValue)&gt;
&lt;!ELEMENT Delete (QualifiedTag+)&gt;
&lt;!ELEMENT QualifiedTag (#PCDATA)&gt;
&lt;!ELEMENT MetadataValue (#PCDATA)&gt;
&lt;!ELEMENT ProviderLastUpdateDateTime (#PCDATA)&gt;
&lt;!ELEMENT SaveDateTimeFlag (SAVE | DONTSAVE)&gt;
&lt;!ELEMENT SAVE EMPTY&gt;
&lt;!ELEMENT ID (#PCDATA)&gt;
&lt;!ELEMENT DONTSAVE EMPTY&gt;</pre>
```

---

#### Code Listing 65. BMGT Metadata Update DTD

---

```
<!ELEMENT ProviderAccountService (UpdateMetadata)>
```

```
<!--UpdateMetadata can update a single collection, multiple collections, a
single granule, or multiple granules in one transaction. Each update allows the
addition of new metadata-->
<!ELEMENT UpdateMetadata (Collection*, Granule*)>
<!ELEMENT Collection (Target+, (Add | Update | Delete)+)>
<!ELEMENT Granule (Target+, (Add | Update | Delete)+)>
<!-- Target+ allows the same change to be made to several different granules or
collections simultaneously. This is especially useful for bulk deletions of
OnlineURLs. -->
<!ELEMENT Target (ID, ProviderLastUpdateDateTime, SaveDateTimeFlag?)>
<!-- SaveDateTimeFlag is the flag that allows echo to update the last update
date time for Target. The default is SAVE -->
<!ELEMENT Add (QualifiedTag, MetadataValue)>
<!ELEMENT Update (QualifiedTag, MetadataValue)>
<!ELEMENT Delete (QualifiedTag+)>
<!ELEMENT QualifiedTag (#PCDATA)>
<!ELEMENT MetadataValue (#PCDATA)>
<!ELEMENT ProviderLastUpdateDateTime (#PCDATA)>
<!ELEMENT SaveDateTimeFlag (SAVE | DONTSAVE)>
<!ELEMENT SAVE EMPTY>
<!ELEMENT ID (#PCDATA)>
<!ELEMENT DONTSAVE EMPTY>
```

---

#### Code Listing 66. Additional Attributes Element in Metadata Model

---

```
<!ELEMENT AdditionalAttributes (AdditionalAttributeDataType,
AdditionalAttributeDescription, AdditionalAttributeName, MeasurementResolution?,
ParameterRangeBegin?, ParameterRangeEnd?, ParameterUnitsOfMeasure?,
ParameterValueAccuracy?, ValueAccuracyExplanation?, ParameterValue*)>
<!ELEMENT AdditionalAttributeDataType (#PCDATA)>
<!ELEMENT AdditionalAttributeDescription (#PCDATA)>
<!ELEMENT AdditionalAttributeName (#PCDATA)>
```

The **QualifiedTag** is a string that follows the XPath standard. (XPath is a language for finding information in an XML document and navigating through elements and attributes.) It is used to indicate which item(s) in the ECHO metadata DTD are to be updated, inserted, or deleted. When used to delete all granule-level Online Access URLs for granules or collections, the XPath needs to use **OnlineAccessURLs**. An optional flag, **SaveDateTimeFlag**, allows you to change the date/time timestamp that identifies when metadata was last updated, to reflect new update date/time information, and the **ScienceQualityFlag** allow you to indicate the quality level a specific granule's parameter. The following are examples of metadata updates

---

#### Code Listing 67: Removing All Online Data URLs for a Granule without Updating the Last Update for Granules Gr1 and Gr2

---

```
<ProviderAccountService>
<UpdateMetadata>
<Granule>
<Target>
<ID>Gr1</ID>
<ProviderLastUpdateDateTime>
2002-06-03 23:00:00
</ProviderLastUpdateDateTime>
<SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
</Target>
<Target>
```

```
<ID>Gr2</ID>
<ProviderLastUpdateDateTime>
2002-06-03 23:00:00
</ProviderLastUpdateDateTime>
<SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
</Target>
<Delete>
<QualifiedTag>OnlineAccessURLs</QualifiedTag>
</Delete>
</Granule>
</UpdateMetadata>
</ProviderAccountService>
```

---

**Code Listing 68: Deleting a Specific Online Data URL**

---

```
<ProviderAccountService>
<UpdateMetadata>
<Granule>
<Target>
<ID>Gr1</ID>
<ProviderLastUpdateDateTime>
2002-06-03 23:00:00
</ProviderLastUpdateDateTime>
<SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
</Target>
<Delete>
<QualifiedTag>
OnlineAccessURLs/OnlineAccessURL [URL="http://jp.provider.com/Gr1URL"]
</QualifiedTag>
</Delete>
</Granule>
</UpdateMetadata>
</ProviderAccountService>
```

---

**Code Listing 69: Insert a New Online Resource**

---

```
<ProviderAccountService>
<UpdateMetadata>
<Granule>
<Target>
<ID>Gr1</ID>
<ProviderLastUpdateDateTime>
2002-06-03 23:00:00
</ProviderLastUpdateDateTime>
<SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
</Target>
<Add>
<QualifiedTag>GranuleOnlineResources/OnlineResource/OnlineResourceURL
</QualifiedTag>
<MetadataValue>http://jp.provider.com/Gr1URL.metadata</MetadataValue>
</Add>
<Add>
<QualifiedTag>
```

```
GranuleOnlineResources/OnlineResource[OnlineResourceURL="http://jp.provider.com/
Gr1URL.metadata"]/OnlineResourceDescription
</QualifiedTag>
<MetadataValue>Metadata cashed in Japan</MetadataValue>
</Add>
<Add>
<QualifiedTag>
GranuleOnlineResources/OnlineResource[OnlineResourceURL="http://jp.provider.com/
Gr1URL.metadata"]/OnlineResourceType
</QualifiedTag>
<MetadataValue>Metadata</MetadataValue>
</Add>
<Add>
<QualifiedTag>
GranuleOnlineResources/OnlineResource[OnlineResourceURL="http://jp.provider.com/
Gr1URL.metadata"]/OnlineResourceMimeType
</QualifiedTag>
<MetadataValue>text</MetadataValue>
</Add>
</Granule>
</UpdateMetadata>
</ProviderAccountService>
```

---

**Code Listing 70: Update Online Data**

---

```
<ProviderAccountService>
<UpdateMetadata>
<Granule>
<Target>
<ID>Gr1</ID>
<ProviderLastUpdateDateTime>
2002-06-03 23:00:00
</ProviderLastUpdateDateTime>
<SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
</Target>
<Update>
<QualifiedTag>
OnlineAccessURL[URL="http://jp.provider.com/Gr1URL"]/URL
</QualifiedTag>
<MetadataValue>http://jp.provider.com/GRANULES/Gr1URL</MetadataValue>
</Update>
</Granule>
</UpdateMetadata>
</ProviderAccountService>
<ProviderAccountService>
<UpdateMetadata>
<Granule>
<Target>
<ID>Gr1</ID>
<ProviderLastUpdateDateTime>
2002-06-03 23:00:00
</ProviderLastUpdateDateTime>
<SaveDateTimeFlag><SAVE/></SaveDateTimeFlag>
</Target>
<Update>
```

```
<QualifiedTag>MeasuredParameter/MeasuredParameterContainer
[ParameterName="Param1"] /QAFlags/AutomaticQualityFlag
</QualifiedTag>
<MetadataValue>automatic quality flag</MetadataValue>
</Update>
<Update>
<QualifiedTag>MeasuredParameter/MeasuredParameterContainer
[ParameterName="Param1"] /QAFlags/AutomaticQualityFlagExplanation
</QualifiedTag>
<MetadataValue>automatic quality flag explanation</MetadataValue>
</Update>
</Granule>
</UpdateMetadata>
</ProviderAccountService>
```

## C.2 BMGT & ECHO 8/9 PARTIAL METADATA UPDATES/ADDS AND DELETES

BMGT and ECHO 8/9 Partial Metadata Updates/adds/deletes includes both collections and granules but only includes the following elements:

a. Partial Metadata Updates/Add

- 1) Collections
  - a) Restriction Flag
  - b) Browse Product/Browse Granule Id
  - c) Visibility ('true' or 'false' without the quotes)
- 2) Granules
  - a) Online access URL
  - b) Online resource URL
  - c) Measured Parameter/Science Quality Flag

b. Deletes

- 1) Collections
  - a) Restriction Flag
  - b) Browse Product/Browse Granule Id

### C.2.1 Delete Items

Instruct ECHO to delete collections or granules by placing the collections or granules under the `<DeleteCollections>` or `<DeleteGranules>` tag, respectively. ECHO uses only the item's identification to process the deletion of the item and all the metadata associated with this item. ECHO will keep the deleted items' identification and deletion date in the database for metadata history auditing purposes. The only way to re-install the collections or granules in the ECHO system is to re-submit the metadata for those items to ECHO for insertion.

### C.2.2 Values for the Qualified Tag for Updates, Deletes, and Inserts

The following DTD excerpts list the acceptable tags for indicating which field in the ECHO data model to update.

*Note: Only the following fields in the data model are available for partial update:  
OnlineResourceURL, OnlineAccessURL, and ScienceQualityField.*

**Code Listing 71. DTD Excerpt for Collection Metadata**


---

```
<!ELEMENT OnlineAccessURLs (OnlineAccessURL +)>
<!ELEMENT OnlineAccessURL (URL, URLDescription?,MimeType)>
<!ELEMENT CollectionOnlineResources (OnlineResource +)>
<!ELEMENT OnlineResource (OnlineResourceURL, OnlineResourceDescription?, 
OnlineResourceType, OnlineResourceMimeType)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT URLDescription (#PCDATA)>
<!ELEMENT MimeType (#PCDATA)>
<!ELEMENT OnlineResourceURL(#PCDATA)>
<!ELEMENT OnlineResourceDescription(#PCDATA)>
<!ELEMENT OnlineResourceType (#PCDATA)>
<!ELEMENT OnlineResourceMimeType (#PCDATA)>
```

---

**Code Listing 72. DTD Excerpt for Granule Metadata**


---

```
<!ELEMENT OnlineAccessURLs (OnlineAccessURL +)>
<!ELEMENT OnlineAccessURL (URL, URLDescription?,MimeType)>
<!ELEMENT GranuleOnlineResources (OnlineResource +)>
<!ELEMENT OnlineResource (OnlineResourceURL, OnlineResourceDescription?, 
OnlineResourceType, OnlineResourceMimeType)>
<!ELEMENT URL (#PCDATA)>
<!ELEMENT URLDescription (#PCDATA)>
<!ELEMENT MimeType (#PCDATA)>
<!ELEMENT OnlineResourceURL(#PCDATA)>
<!ELEMENT OnlineResourceDescription(#PCDATA)>
<!ELEMENT OnlineResourceType (#PCDATA)>
<!ELEMENT OnlineResourceMimeType (#PCDATA)>
```

---

Use OnlineAccessURLs only for the actual data. For URLs to all other kinds of Web pages (including metadata), send these as OnlineResources.

This table identifies valid XPath values for the qualified tags.

**Table 3. Valid XPath Values for the Qualified Tags**

<b>Tag</b>	<b>Function</b>	<b>XPath</b>	<b>Value</b>
Granule URL	Delete One	OnlineAccessURLs/OnlineAccessURL[URL="old_url"]/URL	
Granule URL	Delete All	OnlineAccessURLs	
Granule URL	Update	OnlineAccessURLs /OnlineAccessURL[URL="old_url"]/URL	New URL
Granule URL	Insert	OnlineAccessURLs /OnlineAccessURL/URL url	URL
Granule Online Resource	Delete One	GranuleOnlineResources/OnlineResource[OnlineResourceURL="old_url"] / OnlineResourceURL	
Online Resource URL	Update	GranuleOnlineResources/OnlineResource[OnlineResourceURL="old_url"] /OnlineResourceURL	New URL
Granule Online Resource Type	Update	GranuleOnlineResources/OnlineResource[OnlineResourceURL = "old_url"]/OnlineResourceType	New type

Tag	Function	XPath	Value
Granule Online Resource Mime Type	Insert	GranuleOnlineResources/OnlineResource[OnlineResourceURL = "old_url"]/OnlineResourceMimeType	Mime type
Granule Automatic QA Flag	Update	MeasuredParameter/MeasuredParameterContainer [ParameterName="p1"]/QAFlags/AutomaticQualityFlag	New flag
Granule Automatic QA Flag	Update	MeasuredParameter/MeasuredParameterContainer [ParameterName="p1"]/QAFlags/AutomaticQualityFlagExplanation	New expl.
Granule Operational QA Flag	Update	MeasuredParameter/MeasuredParameterContainer [ParameterName="p1"]/QAFlags/OperationalQualityFlag	New flag
Granule Operational QA Flag Explanation	Update	MeasuredParameter/MeasuredParameterContainer [ParameterName="p1"]/QAFlags/OperationalQualityFlagExplanation	New expl.
Granule Science QA Flag	Update	MeasuredParameter/MeasuredParameterContainer [ParameterName="p1"]/QAFlags/ScienceQualityFlag	New flag
Granule Science QA Flag Explanation	Update	MeasuredParameter/MeasuredParameterContainer [ParameterName="p1"]/QAFlags/ScienceQualityFlagExplanation	New expl.
Additional Attribute	Delete One	AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName= "old_name"]/AdditionalAttributeName	
Additional Attributes	Delete All	AdditionalAttributes	
Additional Attributes	Update	AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName= "old_name"]/AdditionalAttributeName	New name
		AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName= "old_name"]/AdditionalAttributeValue	New value
		AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName= "old_name"]/AdditionalAttributeValueType	New type
Additional Attributes	Insert	AdditionalAttributes/AdditionalAttribute/AdditionalAttributeName	name
		AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName= "name"]/AdditionalAttributeValue	value
		AdditionalAttributes/AdditionalAttribute[AdditionalAttributeName= "name"]/AdditionalAttributeValueType	Type
Visibility	Update	Visible=true   false	Value

## C.2.3 Examples

### C.2.3.1 Partial Metadata Insert

#### Code Listing 73. Partial Metadata Insert

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAccountService>
    <UpdateMetadata>
        <Granule>
```

```
<Target>
    <ID>SC:MIL2TCAL.002:17872867</ID>
    <ProviderLastUpdateDateTime>2007-04-12
00:02:46.95</ProviderLastUpdateDateTime>
    <SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
</Target>
<Add>
<QualifiedTag>GranuleOnlineResources/OnlineResource/OnlineResourceURL</Qualified
Tag>
<MetadataValue>ftp://10dps01u.ecs.nasa.gov/distribution/BRWS/Browse.001/2007.04.
03/MISBR.A2007093.0137.005.2007101133056.AN.24.hdf</MetadataValue>
    </Add>
    <Add>
<QualifiedTag>GranuleOnlineResources/OnlineResource[OnlineResourceURL="ftp://10d
ps01u.ecs.nasa.gov/distribution/BRWS/Browse.001/2007.04.03/MISBR.A2007093.0137.0
05.2007101133056.AN.24.hdf"]/OnlineResourceType</QualifiedTag>
    <MetadataValue>BROWSE</MetadataValue>
    </Add>
    <Add>
<QualifiedTag>GranuleOnlineResources/OnlineResource[OnlineResourceURL="ftp://10d
ps01u.ecs.nasa.gov/distribution/BRWS/Browse.001/2007.04.03/MISBR.A2007093.0137.0
05.2007101133056.AN.24.hdf"]/OnlineResourceMimeType</QualifiedTag>
    <MetadataValue>image/jpeg</MetadataValue>
    </Add>
    <Add>
<QualifiedTag>OnlineAccessURLs/OnlineAccessURL/URL</QualifiedTag>
<MetadataValue>ftp://10dps01u.ecs.nasa.gov/longterm/MISR/MIL2TCAL.002/2007.04.03
/MISR_AM1_TC_ALBEDO_P114_0038776_F04_0008.hdf</MetadataValue>
    </Add>
    <Add>
<QualifiedTag>OnlineAccessURLs/OnlineAccessURL [URL="ftp://10dps01u.ecs.nasa.gov/
longterm/MISR/MIL2TCAL.002/2007.04.03/MISR_AM1_TC_ALBEDO_P114_0038776_F04_0008.h
df"]/MimeType</QualifiedTag>
    <MetadataValue>application/x-hdfeoss</MetadataValue>
    </Add>
    <Add>

<QualifiedTag>GranuleOnlineResources/OnlineResource/OnlineResourceURL</Qualified
Tag>
<MetadataValue>ftp://10dps01u.ecs.nasa.gov/longterm/MISR/MIL2TCAL.002/2007.04.03
/MISR_AM1_TC_ALBEDO_P114_0038776_F04_0008.hdf.xml</MetadataValue>
    </Add>
    <Add>
<QualifiedTag>GranuleOnlineResources/OnlineResource[OnlineResourceURL="ftp://10d
ps01u.ecs.nasa.gov/longterm/MISR/MIL2TCAL.002/2007.04.03/MISR_AM1_TC_ALBEDO_P114
_0038776_F04_0008.hdf.xml"]/OnlineResourceType</QualifiedTag>
    <MetadataValue>METADATA</MetadataValue>
    </Add>
    <Add>
<QualifiedTag>GranuleOnlineResources/OnlineResource[OnlineResourceURL="ftp://10d
ps01u.ecs.nasa.gov/longterm/MISR/MIL2TCAL.002/2007.04.03/MISR_AM1_TC_ALBEDO_P114
_0038776_F04_0008.hdf.xml"]/OnlineResourceMimeType</QualifiedTag>
    <MetadataValue>text/xml</MetadataValue>
    </Add>
    </Granule>
</UpdateMetadata>
```

---

```
</ProviderAccountService>
```

---

### C.2.3.2 Partial Metadata Delete

#### Code Listing 74. Partial Metadata Delete

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAccountService>
    <UpdateMetadata>
        <Granule>
            <Target>
                <ID>SC:MIL3DAE.004:17176039</ID>
                <ProviderLastUpdateDateTime>2007/04/23
08:32:56</ProviderLastUpdateDateTime>
                <SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
            </Target>
            <Target>
                <ID>SC:MIL3DLS.004:17176041</ID>
                <ProviderLastUpdateDateTime>2007/04/23
08:32:56</ProviderLastUpdateDateTime>
                <SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
            </Target>
            <Target>
                <ID>SC:MIL3DAE.004:17176328</ID>
                <ProviderLastUpdateDateTime>2007/04/23
08:32:56</ProviderLastUpdateDateTime>
                <SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
            </Target>
            <Target>
                <ID>SC:MIL3DLS.004:17176329</ID>
                <ProviderLastUpdateDateTime>2007/04/23
08:32:56</ProviderLastUpdateDateTime>
                <SaveDateTimeFlag><DONTSAVE/></SaveDateTimeFlag>
            </Target>
            <Delete>
                <QualifiedTag>OnlineAccessURLs</QualifiedTag>
            <QualifiedTag>GranuleOnlineResources</QualifiedTag>
                </Delete>
            </Granule>
        </UpdateMetadata>
    </ProviderAccountService>
```

---

### C.2.3.3 Partial Granule Update

#### Code Listing 75. Partial Granule Update

---

```
<?xml version="1.0" encoding="UTF-8"?>
<ProviderAccountService>
<UpdateMetadata>
<Granule>
    <Target>
        <ID>SC:MOD14.004:2034404150</ID>
        <ProviderLastUpdateDateTime>2007-04-13
12:07:09.603</ProviderLastUpdateDateTime>
    </Target>
```

```
<Update>
<QualifiedTag>MeasuredParameter/MeasuredParameterContainer[ParameterName="MODIS
L2 Active Fire Detection"]/QAFlags/ScienceQualityFlag</QualifiedTag>
    <MetadataValue>Being Investigated</MetadataValue>
</Update>
</Granule>
<Granule>
    <Target>
        <ID>SC:MOD14.004:2034404150</ID>
        <ProviderLastUpdateDateTime>2007-04-13
12:07:09.603</ProviderLastUpdateDateTime>
    </Target>
    <Update>
<QualifiedTag>MeasuredParameter/MeasuredParameterContainer[ParameterName="MODIS
L2 Active Fire Detection"]/QAFlags/ScienceQualityFlagExplanation</QualifiedTag>
    <MetadataValue>Product assessment is ongoing</MetadataValue>
</Update>
</Granule>
</UpdateMetadata>
</ProviderAccountService>
```

---

## C.2.4 Browse Insert, Update, Replace and Delete

ECHO allows you to update/replace and delete Browse references. Collection and granule metadata associated with the Browse metadata to be ingested must be in the ECHO database.

Browse metadata for ingest must conform to the Browse DTD as shown:

**Code Listing 76. Browse DTD**

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XMLSpy v2005 rel. 3 U (http://www.altova.com)--&gt;
&lt;!ELEMENT BrowseReferenceFile (DTDVersion, DataCenterId, TemporalCoverage,
DeleteBrowse*, BrowseCrossReference*)&gt;
&lt;!ELEMENT DTDVersion (#PCDATA)&gt;
&lt;!ELEMENT DataCenterId (#PCDATA)&gt;
&lt;!ELEMENT TemporalCoverage (StartDate, EndDate)&gt;
&lt;!ELEMENT StartDate (#PCDATA)&gt;
&lt;!ELEMENT EndDate (#PCDATA)&gt;
&lt;!ELEMENT DeleteBrowse (((ShortName, VersionID) | DataSetID | GranuleUR),
InternalFileName*, BrowseCollectionId?,
BrowseGranuleId?)&gt;
&lt;!ATTLIST DeleteBrowse
SilentDeleteErrorHandling NMTOKEN "0"&gt;
&lt;!ELEMENT ShortName (#PCDATA)&gt;
&lt;!ELEMENT VersionID (#PCDATA)&gt;
&lt;!ELEMENT DataSetID (#PCDATA)&gt;
&lt;!ELEMENT GranuleUR (#PCDATA)&gt;
&lt;!ELEMENT InternalFileName (#PCDATA)&gt;
&lt;!ELEMENT BrowseCollectionId (#PCDATA)&gt;
&lt;!ELEMENT BrowseGranuleId (#PCDATA)&gt;
&lt;!ELEMENT BrowseCrossReference (((ShortName, VersionID) | DataSetID |
GranuleUR), BrowseCollectionId?, BrowseGranuleId?,
InsertTime?, LastUpdate?, DeleteTime?, InternalFileName, BrowseDescription?,
BrowseSize?)&gt;
&lt;!ELEMENT InsertTime (#PCDATA)&gt;</pre>
```

```
<!ELEMENT LastUpdate (#PCDATA)>
<!ELEMENT DeleteTime (#PCDATA)>
<!ELEMENT BrowseDescription (#PCDATA)>
<!ELEMENT BrowseSize (#PCDATA)>
```

The following table defines the elements used in Browse ingest insert/update/delete:

**Table 4. Browse Elements**

Element	Definition
DTDVersion	This element specifies the version of the Browse DTD, for example, 9.0. Currently, this field is not captured by ingest.
DataCenterId	The DataCenterID of the data center from which the data is originating. Note: This field is not captured by ingest.
GranuleUR	The GranuleUR for the granule associated with the Browse file.
StartDate	Temporal start date of the data. Note: Currently is not captured by ingest.
EndDate	Temporal end date of the data. Note: currently is not captured by ingest.
BrowseCrossReference	The tag needed for inserting or updating Browse files
DeleteBrowse	The tag needed for deleting Browse files
InternalFileName	The name of the Browse image you will be associating with the granule.
InsertTime	Insert time that the granule was inserted into ECHO.
LastUpdate	The LocalLastUpdate time for the granule in ECHO. The new LastUpdate date must be more recent than old LastUpdate date when updating Browse.
DeleteTime	The time to delete the Browse from the system. If DeleteTime tag not used, then deletion occurs instantaneously.
BrowseSize	The size of the Browse file being inserted or updated for a granule. The value in the Browse ingest file MUST match the actual file size.
SilentDeleteErrorHandler	=0: will return error message if any needed item (e.g. ShortName, VersionID, DatasetID, or GranuleUR) does not exist. =1: turns off error messages

The following DTDs can also be downloaded from the Reference page of the ECHO website at <http://www.echo.nasa.gov/reference/reference.shtml>.

#### C.2.4.1 Insert Browse Example

The code listing below is an XML example of inserting Browse references into multiple granules. The sample XML shows inserting multiple Browse image files and associated with granules. This sample will associate a unique Browse image file with each granule.

**Code Listing 77. Sample Insert Browse XML File**

```
<?xml version="1.0" encoding="UTF-8"?>
<BrowseReferenceFile>
<DTDVersion>1.0</DTDVersion>
<DataCenterId>EDC</DataCenterId>
<TemporalCoverage>
<StartDate>2005-05-12T00:00:00.000Z</StartDate>
<EndDate>2005-06-12T00:00:00.000Z</EndDate>
</TemporalCoverage>
<BrowseCrossReference>
```

```
<GranuleUR>10x10ReferenceGranule0:0</GranuleUR>
<LastUpdate>2005-05-12 07:13:57.05</LastUpdate>
<InternalFileName>test.hdf0</InternalFileName>
<BrowseSize>10</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
<GranuleUR>10x10ReferenceGranule0:1</GranuleUR>
<LastUpdate>2005-05-12 06:47:12.383</LastUpdate>
<InternalFileName>test.hdf1</InternalFileName>
<BrowseSize>11</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
<GranuleUR>10x10ReferenceGranule0:2</GranuleUR>
<LastUpdate>2005-05-12 06:30:03.576</LastUpdate>
<InternalFileName>test.hdf2</InternalFileName>
<BrowseSize>12</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
<GranuleUR>10x10ReferenceGranule0:3</GranuleUR>
<LastUpdate>2005-05-12 06:22:19.323</LastUpdate>
<InternalFileName>test.hdf3</InternalFileName>
<BrowseSize>13</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
<GranuleUR>10x10ReferenceGranule0:4</GranuleUR>
<LastUpdate>2005-05-12 05:23:35.37</LastUpdate>
<InternalFileName>test.hdf4</InternalFileName>
<BrowseSize>14</BrowseSize>
</BrowseCrossReference>
</BrowseReferenceFile>
```

---

To submit the Browse metadata for ingest, place the file and the corresponding Browse image files to the your FTP location.

As shown above, to associate Browse metadata to an existing collection/granule already in the database, include the collection or granule identifier as part of the BrowseCrossReference information.

#### C.2.4.2 Update Browse Example

The following code listing is an example of updating Browse references in multiple granules. The sample XML shows updating the Browse references for the five granules shown in the previous code listing.

##### Code Listing 78. Update Browse Example: testbrowseu.xml

---

```
<?xml version="1.0" encoding="UTF-8"?>
<BrowseReferenceFile>
    <DTDVersion>1.0</DTDVersion>
    <DataCenterId>EDC</DataCenterId>
    <TemporalCoverage>
        <StartDate>2005-05-12T00:00:00.000Z</StartDate>
        <EndDate>2005-06-12T00:00:00.000Z</EndDate>
    </TemporalCoverage>
    <BrowseCrossReference>
        <GranuleUR>10x10ReferenceGranule0:0</GranuleUR>
        <LastUpdate>2005-05-16 07:13:57.05</LastUpdate>
        <InternalFileName>test.hdf1</InternalFileName>
        <BrowseSize>11</BrowseSize>
```

```
</BrowseCrossReference>
<BrowseCrossReference>
  <GranuleUR>10x10ReferenceGranule0:1</GranuleUR>
  <LastUpdate>2005-05-16 06:47:12.383</LastUpdate>
  <InternalFileName>test.hdf2</InternalFileName>
  <BrowseSize>12</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
  <GranuleUR>10x10ReferenceGranule0:2</GranuleUR>
  <LastUpdate>2005-05-16 06:30:03.576</LastUpdate>
  <InternalFileName>test.hdf3</InternalFileName>
  <BrowseSize>13</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
  <GranuleUR>10x10ReferenceGranule0:3</GranuleUR>
  <LastUpdate>2005-05-16 06:22:19.323</LastUpdate>
  <InternalFileName>test.hdf4</InternalFileName>
  <BrowseSize>14</BrowseSize>
</BrowseCrossReference>
<BrowseCrossReference>
  <GranuleUR>10x10ReferenceGranule0:4</GranuleUR>
  <LastUpdate>2005-05-16 05:23:35.37</LastUpdate>
  <InternalFileName>test.hdf0</InternalFileName>
  <BrowseSize>10</BrowseSize>
</BrowseCrossReference>
</BrowseReferenceFile>
```

*NOTE: For the Browse updates to be successful, the file name must be the same as the insert filename.*

*Example: If you associated a Browse image with a granule, and your ingest file was named testbrowse.xml, then your update file must also be named testbrowse.xml. If the file names are not the same, ECHO Ingest will treat the input file as an insert instead of an update.*

To update Browse items, place the Browse update file and the corresponding browse image files to your FTP location.

#### C.2.4.3 Delete Browse Example

The following code listing is an example of deleting Browse references. The sample XML shows the removal of the Browse references previously inserted and updated in the two previous code listings.

##### Code Listing 79. Delete Browse Example: testbrowsed.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<BrowseReferenceFile>
  <DTDVersion>1.0</DTDVersion>
  <DataCenterId>EDC</DataCenterId>
  <TemporalCoverage>
    <StartDate>2005-04-11T00:00:00.000Z</StartDate>
    <EndDate>2005-04-12T00:00:00.000Z</EndDate>
  </TemporalCoverage>
  <DeleteBrowse SilentDeleteErrorHandler = "0">
```

```
<GranuleUR>10x10ReferenceGranule0:0</GranuleUR>
<InternalFileName>test.hdf1</InternalFileName>
</DeleteBrowse>
<DeleteBrowse SilentDeleteErrorHandler = "0">
<GranuleUR>10x10ReferenceGranule0:1</GranuleUR>
<InternalFileName>test.hdf2</InternalFileName>
</DeleteBrowse>
<DeleteBrowse SilentDeleteErrorHandler = "0">
<GranuleUR>10x10ReferenceGranule0:2</GranuleUR>
<InternalFileName>test.hdf3</InternalFileName>
</DeleteBrowse>
<DeleteBrowse SilentDeleteErrorHandler = "0">
<GranuleUR>10x10ReferenceGranule0:3</GranuleUR>
<InternalFileName>test.hdf4</InternalFileName>
</DeleteBrowse>
<DeleteBrowse SilentDeleteErrorHandler = "0">
<GranuleUR>10x10ReferenceGranule0:4</GranuleUR>
<InternalFileName>test.hdf0</InternalFileName>
</DeleteBrowse>
</BrowseReferenceFile>
```

---

To delete browse image metadata from ECHO, place a Browse delete file to your FTP location.

## APPENDIX D. ECHO PATH URIS

ECHO Path URI is the name given to a specific URI which maps into the metadata of a granule or collection. ECHO paths are based on XPath, a language for finding information in an XML document and navigating through elements and attributes. Refer to section **Error! Reference source not found.**, **Error! Reference source not found.** and **Error! Reference source not found.** for information about using XPath as part of the metadata update process.

### D.1 FORMAT

The general format for the URI is `<echoItemType>://<echoHost>/<echoItemId>[/xpath]`

- a. **echoItemType** := granule (or collection)—This is the syntax to indicate whether the URI is pointing to a granule or collections metadata.
- b. **echoHost** is the address of the ECHO server the metadata is on. This should be left blank for now, as this feature is not currently supported.
- c. **echoItemId** is the item ID of the of the granule or collection (for example, C14016455-PSATEST).
- d. **xpath** is optional. It is an XPath statement that maps into the XML payload returned from a GetMetadata request using the echoItemId.

Here is an example in the correct format. Note that the echoHost is left blank.

```
collection:///C14016455-
PSATEST/%2Fresults%2Fprovider%2Fresult%2FCollectionMetaData%2FECHOItemId%2Ftext%28%29
```

The XPath above is escaped to be put in the URI. The un-escaped format looks like:  
"/results/provider/result/CollectionMetaData/ECHOItemId/text()"

### D.2 BEHAVIOR

The granule mapping URI will retrieve data from the XML metadata of a granule or collection. The behavior that appears depending upon the state of the XPath selection appears below.

**Table 5. ECHO Path URI Behavior**

XPath Selection	Granule	Collection
No XPath included	Returns the entire metadata XML	Returns the entire metadata XML
XPath selects multiple nodes	Returns XML fragment representing multiple nodes	Returns XML fragment representing multiple nodes
XPath selects a node with child nodes and attributes	Returns XML fragment representing node and child nodes	Returns XML fragment representing node and child nodes
XPath selects a node with a single value	Returns single value	Returns single value

## APPENDIX E. ECHO ERROR HANDLING

The ECHO 10.0 Web Service API has advanced error-reporting capabilities. There are 12 types of faults reported by ECHO. They are:

- a. **AuthorizationFault** – Reported by ECHO when a user is not authorized to invoke an operation
- b. **DataSizeLimitFault** – Reported by ECHO to indicate that the data size limit has been exceeded
- c. **DuplicateIdFault** – Reported by ECHO to indicate that an entity with the same ID exists in ECHO already
- d. **InternalFault** – Reported by ECHO when an internal error occurs
- e. **InvalidArgumentException** – Reported to indicate that one or more arguments passed were invalid
- f. **InvalidStateFault** – Reported to indicate that an action by the client would put an object in ECHO in an invalid state
- g. **InvalidURLFault** – Reported to indicate invalid syntax in a URL or an element of the URL that does not exist
- h. **ItemNotFoundFault** – Reported when the client attempts to access one or more objects that do not exist
- i. **ParseFault** – Reported to indicate that some value could not be parsed
- j. **RemovalFault** - Reported to indicate an error that has occurred during the removal of an object from ECHO
- k. **UnsupportedFeatureFault** - Reported to indicated that a feature was selected that is not supported
- l. **ValidationFault** - Reported to indicate that an object in or passed to ECHO is not valid

All the above fault types extend the basic **EchoFault** type. An **ECHOFault** is reported by ECHO when an error occurred during the invocation of an ECHO operation

All faults will include an **ErrorCode**, **SystemMessage**, and **Timestamp** of when the error occurred and an **ErrorInstanceld**. An **EchoFault** may also have an **OpsMessage** (message specified by ECHO Operations).

Error codes are strings that uniquely identify an error case in ECHO. Some error codes are reused, such as when a required parameter to an operation was not provided. ECHO Operations may associate different messages with specific error codes. If ECHO Operations has a message configured for an error code, then that message will be returned with the **EchoFault** in the **OpsMessage** element.

In most instances, receiving a fault from ECHO occurs by catching an **EchoFault** and displaying the Ops Message, System Message, and Error Instance ID to the user.

**InternalFaults** capture errors that are internal to ECHO, such as when ECHO cannot talk to the metadata catalog or when an order attempts to transition to an invalid state. There is nothing a client can do to recover from an **InternalFault** except to report any information provided with the error to ECHO Operations.

### Code Listing 80: Catching Exceptions from ECHO

---

```
try
{
    // Create authentication service
    AuthenticationServiceLocator authServiceLocator =
        new AuthenticationServiceLocator();
    AuthenticationServicePort authenticationService =
        authServiceLocator.getAuthenticationServicePort();

    ClientInformation clientInfo =
        new ClientInformation();
    clientInfo.setClientId("A Client");
    clientInfo.setUserIpAddress("192.168.1.1");
```

---

```
// Call login with jdoe as username, mypass as password,
// and client information
authenticationService.login("jdoe", "mypass",
    clientInfo, null, null);
}
catch (EchoFault e)
{
    // This exception was likely caused by user input.
    String message = "Could not login to ECHO:";

    if (e.getOpsMessage() != null
        && e.getOpsMessage().length() > 0)
    {
        message += "\nOps Message : " + e.getOpsMessage();
    }
    message +=
        "\nMessage: " + e.getSystemMessage()
        + "\nError Instance Id: "
        + e.getErrorInstanceId();
    System.out.println(message);
}
catch (RemoteException e)
{
    // ECHO could not be reached.
    System.out
        .println("Could not communicate with ECHO :"
            + e.toString());
}
catch ( ServiceException e)
{
    // An error occurred while creating the service.
    System.out.println("Could not create ECHO Service :"
        + e.toString());
}
```

---

## APPENDIX F. INGEST SCHEMAS AND DTDS

The following schemas and DTDs can also be downloaded from the ECHO website on the Data Partners Development page at [http://www.echo.nasa.gov/data\\_partners/data-devel.shtml](http://www.echo.nasa.gov/data_partners/data-devel.shtml):

- Collection Metadata
- Granule Metadata
- Update Metadata
- Browse Metadata

Beginning with ECHO 10.0, ingest data is described via schemas. However, to support legacy ECHO, the legacy DTDs remain.

## APPENDIX G. BEST PRACTICES

The following tips and other recommended practices are in outline form rather than paragraphs for quick reading.

### G.1 TIPS FOR PREPARING YOUR METADATA

- a. File naming conventions can help you track a file in the final Ingest report
  - 1) A metadata file name should be self-describing and unique, including ID of data provider, metadata type, short name, and version\_id. The date coverage of metadata and generation date is helpful, for example:  
NSCGAMSR200701620070160101.20070116113342.XML  
EDCBMOLT200701620070170202.20070117005857.XML
- b. Validate before submission
- c. Perform ingest of new metadata on partner test system
- d. There are separate schemas that govern the XML format for the various metadata types, available from the ECHO website at <http://www.echo.nasa.gov>  
Click: Data Partners → Data Development
- e. There are also examples of minimum collection and granule metadata XML requirements located on the ECHO website at <http://www.echo.nasa.gov>  
Click: Data Partners → Getting Started

### G.2 FACTORS THAT AFFECT INGEST RATES

- a. Metadata type being ingested (collection, granule, browse, update)
- b. Type of action being taken (insert, delete, replacement)
- c. Amount of information in a metadata record (sparse vs. dense)
- d. Number of records in a provider schema
- e. Amount of time required for preprocessing such as adapting to the current format
- f. Amount of competition for database and system resources

### G.3 COMMON ERRORS

- a. Use of incorrect data type for element:
  - 1) Use of string for a date or number field:
    - a) <DeleteTime>none</DeleteTime>
    - b) <SwathWidth>2600 km</SwathWidth>
- b. Exceeding the character limitations for an element—refer to entity definitions on ECHO website.
- c. Incomplete dates
- d. Not matching short name and version\_id of granule to what is used with its associated collection.
- e. Violating unique constraints within a collection or granule

## APPENDIX H. INSERT XML EXAMPLES

### H.1 COLLECTION

```
<CollectionMetaDataFile>
<Collections>
<Collection>
<ShortName>TESTCOLL<0</ShortName>
<VersionId>1.0</VersionId>
<InsertTime>2009<01<05T05:30:30.550<05:00</InsertTime>
<LastUpdate>2009<01<05T05:30:30.550<05:00</LastUpdate>
<DeleteTime>2009<01<05T05:30:30.550<05:00</DeleteTime>
<LongName>Test Collection 1 Long Name</LongName>
<DataSetId>Test Collection<0</DataSetId>
<Description>This is a test collection that we use for unit testing out data</Description>
<Orderable>true</Orderable>
<Visible>true</Visible>
<RevisionDate>2009<01<05T05:30:30.550<05:00</RevisionDate>
<SuggestedUsage>This is the suggested usage for this collection</SuggestedUsage>
<ProcessingCenter>Goddard Space Flight Center</ProcessingCenter>
<ProcessingLevelId>Level 4</ProcessingLevelId>
<ProcessingLevelDescription>The forth level of processing</ProcessingLevelDescription>
<ArchiveCenter>GSFCS4PA</ArchiveCenter>
<VersionDescription>Fixed a bug in range calc</VersionDescription>
<CitationForExternalPublication>Contact Bill Smith</CitationForExternalPublication>
<CollectionState>Final</CollectionState>
<MaintenanceAndUpdateFrequency>Once a week</MaintenanceAndUpdateFrequency>
<RestrictionFlag>4.2</RestrictionFlag>
<RestrictionComment>Restricted to level four and above</RestrictionComment>
<Price>32.45</Price>
<DataFormat>JPEG Images</DataFormat>
<SpatialKeywords>
<Keyword>Word<1</Keyword>
<Keyword>Word<0</Keyword>
</SpatialKeywords>
<TemporalKeywords>
<Keyword>Word<3</Keyword>
```

```
<Keyword>Word<2</Keyword>
</TemporalKeywords>
<Temporal>
<TimeType>Universal Time</TimeType>
<DateType>Eastern Daylight</DateType>
<TemporalRangeType>Long Range</TemporalRangeType>
<PrecisionOfSeconds>1</PrecisionOfSeconds>
<EndsAtPresentFlag>true</EndsAtPresentFlag>
<SingleDateTime>2009<01<05T05:30:30.550<05:00</SingleDateTime>
</Temporal>
<Contacts>
<Contact>
<Role>Administrator<4</Role>
<HoursOfService>42</HoursOfService>
<Instructions>Call before 4pm</Instructions>
<OrganizationName>GSFCECS</OrganizationName>
<OrganizationAddresses>
<Address>
<StreetAddress>123 Fake St.</StreetAddress>
<City>Greenbelt</City>
<StateProvince>Maryland</StateProvince>
<PostalCode>20904</PostalCode>
<Country>USA</Country>
</Address>
<Address>
<StreetAddress>123 Fake St.</StreetAddress>
<City>Greenbelt</City>
<StateProvince>Maryland</StateProvince>
<PostalCode>20904</PostalCode>
<Country>USA</Country>
</Address>
</OrganizationAddresses>
<OrganizationPhones>
<Phone>
<Number>555<555<1212</Number>
<Type>Work</Type>
</Phone>
```

```
<Phone>
<Number>555<555<1212</Number>
<Type>Work</Type>
</Phone>
</OrganizationPhones>
<OrganizationEmails>
<Email>tester<5@tester.com</Email>
<Email>tester<6@tester.com</Email>
</OrganizationEmails>
<ContactPersons>
<ContactPerson>
<FirstName>Bob<7</FirstName>
<MiddleName>Robert</MiddleName>
<LastName>Smith</LastName>
<JobPosition>Worker Bee</JobPosition>
</ContactPerson>
<ContactPerson>
<FirstName>Bob<8</FirstName>
<MiddleName>Robert</MiddleName>
<LastName>Smith</LastName>
<JobPosition>Worker Bee</JobPosition>
</ContactPerson>
</ContactPersons>
</Contact>
<Contact>
<Role>Administrator<9</Role>
<HoursOfService>42</HoursOfService>
<Instructions>Call before 4pm</Instructions>
<OrganizationName>GSFCECS</OrganizationName>
<OrganizationAddresses>
<Address>
<StreetAddress>123 Fake St.</StreetAddress>
<City>Greenbelt</City>
<StateProvince>Maryland</StateProvince>
<PostalCode>20904</PostalCode>
<Country>USA</Country>
</Address>
```

```
<Address>
<StreetAddress>123 Fake St.</StreetAddress>
<City>Greenbelt</City>
<StateProvince>Maryland</StateProvince>
<PostalCode>20904</PostalCode>
<Country>USA</Country>
</Address>
</OrganizationAddresses>
<OrganizationPhones>
<Phone>
<Number>555<555<1212</Number>
<Type>Work</Type>
</Phone>
<Phone>
<Number>555<555<1212</Number>
<Type>Work</Type>
</Phone>
</OrganizationPhones>
<OrganizationEmails>
<Email>tester<11@tester.com</Email>
<Email>tester<10@tester.com</Email>
</OrganizationEmails>
<ContactPersons>
<ContactPerson>
<FirstName>Bob<12</FirstName>
<MiddleName>Robert</MiddleName>
<LastName>Smith</LastName>
<JobPosition>Worker Bee</JobPosition>
</ContactPerson>
<ContactPerson>
<FirstName>Bob<13</FirstName>
<MiddleName>Robert</MiddleName>
<LastName>Smith</LastName>
<JobPosition>Worker Bee</JobPosition>
</ContactPerson>
</ContactPersons>
</Contact>
```

```
</Contacts>
<DisciplineTopicParameters>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<14</DisciplineKeyword>
<TopicKeyword>Word<15</TopicKeyword>
<TermKeyword>Word<16</TermKeyword>
<VariableKeyword>Word<17</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<18</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<19</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<20</DisciplineKeyword>
<TopicKeyword>Word<21</TopicKeyword>
<TermKeyword>Word<22</TermKeyword>
<VariableKeyword>Word<23</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<25</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<24</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
</DisciplineTopicParameters>
<Platforms>
<Platform>
<ShortName>Short Name<26</ShortName>
<LongName>Long Name</LongName>
<Type>Platform Type</Type>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<27</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
```

```
<Name>Characteristic Name<28</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Instruments>
<Instrument>
<ShortName>Short Name<29</ShortName>
<LongName>Long Name</LongName>
<Technique>Instrument Technique</Technique>
<NumberOfSensors>10</NumberOfSensors>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<30</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<31</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name<32</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<33</Name>
```

```
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<34</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name<35</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<36</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<37</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
```

```
<OperationMode>Operation Mode<38</OperationMode>
<OperationMode>Operation Mode<39</OperationMode>
</OperationModes>
<DisciplineTopicParameters>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<40</DisciplineKeyword>
<TopicKeyword>Word<41</TopicKeyword>
<TermKeyword>Word<42</TermKeyword>
<VariableKeyword>Word<43</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<45</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<44</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<46</DisciplineKeyword>
<TopicKeyword>Word<47</TopicKeyword>
<TermKeyword>Word<48</TermKeyword>
<VariableKeyword>Word<49</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<51</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<50</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
</DisciplineTopicParameters>
</Instrument>
<Instrument>
<ShortName>Short Name<52</ShortName>
<LongName>Long Name</LongName>
<Technique>Instrument Technique</Technique>
<NumberOfSensors>10</NumberOfSensors>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<53</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
```

```
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<54</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name<55</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<56</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<57</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name<58</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
```

```
<Characteristic>
<Name>Characteristic Name<59</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<60</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode<62</OperationMode>
<OperationMode>Operation Mode<61</OperationMode>
</OperationModes>
<DisciplineTopicParameters>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<63</DisciplineKeyword>
<TopicKeyword>Word<64</TopicKeyword>
<TermKeyword>Word<65</TermKeyword>
<VariableKeyword>Word<66</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<67</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<68</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<69</DisciplineKeyword>
<TopicKeyword>Word<70</TopicKeyword>
<TermKeyword>Word<71</TermKeyword>
<VariableKeyword>Word<72</VariableKeyword>
```

```
<ECSParameterKeywords>
<ECSParameterKeyword>Word<74</ECSParameterKeyword>
<ECSParameterKeyword>Word<73</ECSParameterKeyword>
</ECSParameterKeywords>
</DisciplineTopicParameter>
</DisciplineTopicParameters>
</Instrument>
</Instruments>
</Platform>
<Platform>
<ShortName>Short Name<75</ShortName>
<LongName>Long Name</LongName>
<Type>Platform Type</Type>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<76</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<77</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Instruments>
<Instrument>
<ShortName>Short Name<78</ShortName>
<LongName>Long Name</LongName>
<Technique>Instrument Technique</Technique>
<NumberOfSensors>10</NumberOfSensors>
<Characteristics>
<Characteristic>
```

```
<Name>Characteristic Name<79</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<80</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name<81</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<82</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<83</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
```

```
<ShortName>Short Name<84</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<85</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<86</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode<88</OperationMode>
<OperationMode>Operation Mode<87</OperationMode>
</OperationModes>
<DisciplineTopicParameters>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<89</DisciplineKeyword>
<TopicKeyword>Word<90</TopicKeyword>
<TermKeyword>Word<91</TermKeyword>
<VariableKeyword>Word<92</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<93</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<94</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
<DisciplineTopicParameter>
```

```
<DisciplineKeyword>Word<95</DisciplineKeyword>
<TopicKeyword>Word<96</TopicKeyword>
<TermKeyword>Word<97</TermKeyword>
<VariableKeyword>Word<98</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<99</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<100</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
</DisciplineTopicParameters>
</Instrument>
<Instrument>
<ShortName>Short Name<101</ShortName>
<LongName>Long Name</LongName>
<Technique>Instrument Technique</Technique>
<NumberOfSensors>10</NumberOfSensors>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<102</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<103</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name<104</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
```

```
<Characteristics>
<Characteristic>
<Name>Characteristic Name<105</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<106</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name<107</ShortName>
<LongName>Long Name</LongName>
<Technique>Sensor Technique</Technique>
<Characteristics>
<Characteristic>
<Name>Characteristic Name<108</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name<109</Name>
<Description>Characteristic Description</Description>
<DataType>Data Type</DataType>
<Unit>Characteristic Unit</Unit>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
```

```
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode<110</OperationMode>
<OperationMode>Operation Mode<111</OperationMode>
</OperationModes>
<DisciplineTopicParameters>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<112</DisciplineKeyword>
<TopicKeyword>Word<113</TopicKeyword>
<TermKeyword>Word<114</TermKeyword>
<VariableKeyword>Word<115</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<116</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<117</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
<DisciplineTopicParameter>
<DisciplineKeyword>Word<118</DisciplineKeyword>
<TopicKeyword>Word<119</TopicKeyword>
<TermKeyword>Word<120</TermKeyword>
<VariableKeyword>Word<121</VariableKeyword>
<ECSPParameterKeywords>
<ECSPParameterKeyword>Word<123</ECSPParameterKeyword>
<ECSPParameterKeyword>Word<122</ECSPParameterKeyword>
</ECSPParameterKeywords>
</DisciplineTopicParameter>
</DisciplineTopicParameters>
</Instrument>
</Instruments>
</Platform>
</Platforms>
<AdditionalAttributes>
<AdditionalAttribute>
<Name>Attribute Name<124</Name>
<DataType>Data Type</DataType>
<Description>Attribute Description</Description>
```

```
<MeasurementResolution>Measurement Resolution</MeasurementResolution>
<ParameterRangeBegin>Parameter Range Begin</ParameterRangeBegin>
<ParameterRangeEnd>Parameter Range End</ParameterRangeEnd>
<ParameterUnitsOfMeasure>Parameter Units Of Measure</ParameterUnitsOfMeasure>
<ParameterValueAccuracy>Parameter Value Accuracy</ParameterValueAccuracy>
<ValueAccuracyExplanation>Value Accuracy Explanation</ValueAccuracyExplanation>
<Value>Attribute Value</Value>
</AdditionalAttribute>
<AdditionalAttribute>
<Name>Attribute Name<125</Name>
<DataType>Data Type</DataType>
<Description>Attribute Description</Description>
<MeasurementResolution>Measurement Resolution</MeasurementResolution>
<ParameterRangeBegin>Parameter Range Begin</ParameterRangeBegin>
<ParameterRangeEnd>Parameter Range End</ParameterRangeEnd>
<ParameterUnitsOfMeasure>Parameter Units Of Measure</ParameterUnitsOfMeasure>
<ParameterValueAccuracy>Parameter Value Accuracy</ParameterValueAccuracy>
<ValueAccuracyExplanation>Value Accuracy Explanation</ValueAccuracyExplanation>
<Value>Attribute Value</Value>
</AdditionalAttribute>
<AdditionalAttribute>
<Name>Attribute Name<126</Name>
<DataType>Data Type</DataType>
<Description>Attribute Description</Description>
<MeasurementResolution>Measurement Resolution</MeasurementResolution>
<ParameterRangeBegin>Parameter Range Begin</ParameterRangeBegin>
<ParameterRangeEnd>Parameter Range End</ParameterRangeEnd>
<ParameterUnitsOfMeasure>Parameter Units Of Measure</ParameterUnitsOfMeasure>
<ParameterValueAccuracy>Parameter Value Accuracy</ParameterValueAccuracy>
<ValueAccuracyExplanation>Value Accuracy Explanation</ValueAccuracyExplanation>
<Value>Attribute Value</Value>
</AdditionalAttribute>
</AdditionalAttributes>
SDTDescriptions>
SDTDescription>
<PrimaryCSDT>Earth<127</PrimaryCSDT>
<Implementation>Impl</Implementation>
```

```
<CSDTComments>CSDT Comments</CSDTComments>
<IndirectReference>Indirect Reference</IndirectReference>
</CSDTDescription>
SDTDescription>
<PrimaryCSDT>Earth<128</PrimaryCSDT>
<Implementation>Impl</Implementation>
<CSDTComments>CSDT Comments</CSDTComments>
<IndirectReference>Indirect Reference</IndirectReference>
</CSDTDescription>
</CSDTDescriptions>
<CollectionAssociations>
<CollectionAssociation>
<ShortName>COLLOTHER<129</ShortName>
<VersionId>1</VersionId>
<CollectionType>Input Collection</CollectionType>
<CollectionUse>Extra data</CollectionUse>
</CollectionAssociation>
<CollectionAssociation>
<ShortName>COLLOTHER<130</ShortName>
<VersionId>1</VersionId>
<CollectionType>Input Collection</CollectionType>
<CollectionUse>Extra data</CollectionUse>
</CollectionAssociation>
</CollectionAssociations>
<Campaigns>
<Campaign>
<ShortName>Short Name<131</ShortName>
<LongName>Long Name</LongName>
<StartDate>2009<01<05T05:30:30.550<05:00</StartDate>
<EndDate>2009<01<05T05:30:30.550<05:00</EndDate>
</Campaign>
<Campaign>
<ShortName>Short Name<132</ShortName>
<LongName>Long Name</LongName>
<StartDate>2009<01<05T05:30:30.550<05:00</StartDate>
<EndDate>2009<01<05T05:30:30.550<05:00</EndDate>
</Campaign>
```

```
</Campaigns>
<AlgorithmPackages>
<AlgorithmPackage>
<Name>Alg Name<133</Name>
<Version>1.0</Version>
<Description>Description</Description>
</AlgorithmPackage>
<AlgorithmPackage>
<Name>Alg Name<134</Name>
<Version>1.0</Version>
<Description>Description</Description>
</AlgorithmPackage>
</AlgorithmPackages>
<SpatialInfo>
<SpatialCoverageType>SpatialCoverageType</SpatialCoverageType>
<VerticalCoordinateSystem>
<AltitudeSystemDefinition>
<DatumName>AltitudeDatumName</DatumName>
<DistanceUnits>AltitudeDistanceUnits</DistanceUnits>
<EncodingMethod>AltitudeEncodingMethod</EncodingMethod>
<Resolutions>
<Resolution>0.0</Resolution>
<Resolution>0.1</Resolution>
</Resolutions>
</AltitudeSystemDefinition>
<DepthSystemDefinition>
<DatumName>DepthDatumName</DatumName>
<DistanceUnits>DepthDistanceUnits</DistanceUnits>
<EncodingMethod>DepthEncodingMethod</EncodingMethod>
<Resolutions>
<Resolution>0.0</Resolution>
<Resolution>0.1</Resolution>
</Resolutions>
</DepthSystemDefinition>
</VerticalCoordinateSystem>
<HorizontalCoordinateSystem>
<GeodeticModel>
```

```
<HorizontalDatumName>HorizontalDatumName</HorizontalDatumName>
<EllipsoidName>EllipsoidName</EllipsoidName>
<SemiMajorAxis>2.0</SemiMajorAxis>
<DenominatorOfFlatteningRatio>0.25</DenominatorOfFlatteningRatio>
</GeodeticModel>
<PlanarCoordinateSystems>
<PlanarCoordinateSystem>
<PlanarCoordinateSystemId>Planar Coordinate System Id<135</PlanarCoordinateSystemId>
<PlanarCoordinateInformation>
<DistanceUnits>DistanceUnits</DistanceUnits>
<EncodingMethod>EncodingMethod</EncodingMethod>
<DistanceAndBearingRepresentation>
<DistanceResolution>0.1</DistanceResolution>
<BearingResolution>0.1</BearingResolution>
<BearingUnits>BearingUnits</BearingUnits>
<BearingReferenceDirection>BearingReferenceDir</BearingReferenceDirection>
<BearingReferenceMeridian>BearingReferenceMeridian</BearingReferenceMeridian>
</DistanceAndBearingRepresentation>
</PlanarCoordinateInformation>
<MapProjection>
<MapProjectionName>MapProjectionName</MapProjectionName>
<MapProjectionPointer>MapProjectionPointer</MapProjectionPointer>
</MapProjection>
</PlanarCoordinateSystem>
<PlanarCoordinateSystem>
<PlanarCoordinateSystemId>Planar Coordinate System Id<136</PlanarCoordinateSystemId>
<PlanarCoordinateInformation>
<DistanceUnits>DistanceUnits</DistanceUnits>
<EncodingMethod>EncodingMethod</EncodingMethod>
<CoordinateRepresentation>
<AbscissaResolution>0.1</AbscissaResolution>
<OrdinateResolution>0.1</OrdinateResolution>
</CoordinateRepresentation>
</PlanarCoordinateInformation>
<LocalPlanarCoordinateSystem>
<Description>LocalPlanarCoordinateSystem</Description>
<GeoReferenceInformation>LocalPlanarGeoReferenceInfo</GeoReferenceInformation>
```

```
</LocalPlanarCoordinateSystem>
</PlanarCoordinateSystem>
</PlanarCoordinateSystems>
</HorizontalCoordinateSystem>
</SpatialInfo>
<OnlineAccessURLs>
<OnlineAccessURL>
<URL>http://www.google.com<137</URL>
<URLDescription>URLDescription URLDescription</URLDescription>
<MimeType>MimeTypeMimeType</MimeType>
</OnlineAccessURL>
<OnlineAccessURL>
<URL>http://www.google.com<139</URL>
<URLDescription>URLDescription URLDescription</URLDescription>
<MimeType>MimeTypeMimeType</MimeType>
</OnlineAccessURL>
</OnlineAccessURLs>
<OnlineResources>
<OnlineResource>
<URL>http://www.cnn.com<138</URL>
<Description>Description Description</Description>
<Type>Type Type</Type>
<MimeType>MimeTypeMimeType</MimeType>
</OnlineResource>
<OnlineResource>
<URL>http://www.cnn.com<140</URL>
<Description>Description Description</Description>
<Type>Type Type</Type>
<MimeType>MimeTypeMimeType</MimeType>
</OnlineResource>
</OnlineResources>
<AssociatedDIFs>
<DIF>
<EntryId>EntryID EntryID<141</EntryId>
</DIF>
<DIF>
<EntryId>EntryID EntryID<142</EntryId>
```

```
</DIF>
</AssociatedDIFs>
<Spatial>
<SpatialCoverageType>spatialCoverageType</SpatialCoverageType>
<HorizontalSpatialDomain>
<ZoneIdentifier>zone</ZoneIdentifier>
<Geometry>
<CoordinateSystem>CARTESIAN</CoordinateSystem>
<Point>
<PointLongitude><b>21.4</PointLongitude>
<PointLatitude><b>15.4</PointLatitude>
</Point>
<Point>
<PointLongitude><b>21.4</PointLongitude>
<PointLatitude><b>16.4</PointLatitude>
</Point>
<Point>
<PointLongitude><b>21.4</PointLongitude>
<PointLatitude><b>17.4</PointLatitude>
</Point>
<BoundingRectangle>
<WestBoundingCoordinate><b>163</WestBoundingCoordinate>
<NorthBoundingCoordinate>41</NorthBoundingCoordinate>
<EastBoundingCoordinate><b>156</EastBoundingCoordinate>
<SouthBoundingCoordinate>40</SouthBoundingCoordinate>
</BoundingRectangle>
<BoundingRectangle>
<WestBoundingCoordinate><b>163</WestBoundingCoordinate>
<NorthBoundingCoordinate>39</NorthBoundingCoordinate>
<EastBoundingCoordinate><b>156</EastBoundingCoordinate>
<SouthBoundingCoordinate>38</SouthBoundingCoordinate>
</BoundingRectangle>
<BoundingRectangle>
<WestBoundingCoordinate><b>163</WestBoundingCoordinate>
<NorthBoundingCoordinate>37</NorthBoundingCoordinate>
<EastBoundingCoordinate><b>156</EastBoundingCoordinate>
<SouthBoundingCoordinate>36</SouthBoundingCoordinate>
```

```
</BoundingRectangle>
<GPolygon>
<Boundary>
<Point>
<PointLongitude><b>79.4</b></PointLongitude>
<PointLatitude><b>39.6</b></PointLatitude>
</Point>
<Point>
<PointLongitude><b>75.7</b></PointLongitude>
<PointLatitude><b>39.6</b></PointLatitude>
</Point>
<Point>
<PointLongitude><b>75.2</b></PointLongitude>
<PointLatitude><b>38</b></PointLatitude>
</Point>
<Point>
<PointLongitude><b>77.2</b></PointLongitude>
<PointLatitude><b>38.3</b></PointLatitude>
</Point>
<Point>
<PointLongitude><b>77.5</b></PointLongitude>
<PointLatitude><b>39</b></PointLatitude>
</Point>
<Point>
<PointLongitude><b>79.4</b></PointLongitude>
<PointLatitude><b>39.1</b></PointLatitude>
</Point>
</Boundary>
<ExclusiveZone>
<Boundary>
<Point>
<PointLongitude><b>76.6</b></PointLongitude>
<PointLatitude><b>39.2</b></PointLatitude>
</Point>
<Point>
<PointLongitude><b>76.1</b></PointLongitude>
<PointLatitude><b>39.2</b></PointLatitude>
```

```
</Point>
<Point>
<PointLongitude>76.2</PointLongitude>
<PointLatitude>38.4</PointLatitude>
</Point>
<Point>
<PointLongitude>76.5</PointLongitude>
<PointLatitude>38.4</PointLatitude>
</Point>
</Boundary>
</ExclusiveZone>
</GPolygon>
<Line>
<Point>
<PointLongitude>22.2</PointLongitude>
<PointLatitude>25.8</PointLatitude>
</Point>
<Point>
<PointLongitude>25</PointLongitude>
<PointLatitude>27.7</PointLatitude>
</Point>
<Point>
<PointLongitude>27.3</PointLongitude>
<PointLatitude>25.6</PointLatitude>
</Point>
</Line>
</Geometry>
</HorizontalSpatialDomain>
<VerticalSpatialDomain>
<Type>type1</Type>
<Value>value1</Value>
</VerticalSpatialDomain>
<VerticalSpatialDomain>
<Type>type2</Type>
<Value>value2</Value>
</VerticalSpatialDomain>
<VerticalSpatialDomain>
```

```
<Type>type3</Type>
<Value>value3</Value>
</VerticalSpatialDomain>
<OrbitParameters>
<SwathWidth>13</SwathWidth>
<Period>25</Period>
<InclinationAngle>55</InclinationAngle>
<NumberOfOrbits>1</NumberOfOrbits>
</OrbitParameters>
<GranuleSpatialRepresentation>GEODETIC</GranuleSpatialRepresentation>
</Spatial>
<AssociatedBrowselImages>
<ProviderBrowseld>ProviderBrowseld<0</ProviderBrowseld>
<ProviderBrowseld>ProviderBrowseld<5</ProviderBrowseld>
</AssociatedBrowselImages>
</Collection>
</Collections>
</CollectionMetaDataFile>
```

## H.2 GRANULE

```
<GranuleMetaDataTable>
<Granules>
<Granule>
<GranuleUR>GranuleUR-1</GranuleUR>
<InsertTime>2009-01-05T05:30:30-05:00</InsertTime>
<LastUpdate>2009-01-05T05:30:30-05:00</LastUpdate>
<DeleteTime>2009-01-05T05:30:30-05:00</DeleteTime>
<Collection>
<ShortName>TESTCOLL-0</ShortName>
<VersionId>1.0</VersionId>
</Collection>
<RestrictionFlag>0.0</RestrictionFlag>
<RestrictionComment>RestrictionComment</RestrictionComment>
<DataGranule>
<SizeMBDataGranule>0.0</SizeMBDataGranule>
<ReprocessingPlanned>ReprocessingPlanned</ReprocessingPlanned>
<ReprocessingActual>ReprocessingActual</ReprocessingActual>
<ProducerGranuleId>0000000.0000001.hdf</ProducerGranuleId>
<DayNightFlag>DAY</DayNightFlag>
<ProductionDateTime>2009-01-05T05:30:30-05:00</ProductionDateTime>
<LocalVersionId>LocalVersionId</LocalVersionId>
</DataGranule>
<PGEVersionClass>
<PGENAME>PGENAME</PGENAME>
<PGEVersion>PGEVersion</PGEVersion>
</PGEVersionClass>
<Temporal>
<RangeDateTime>
<BeginningDateTime>2009-01-05T05:30:30-05:00</BeginningDateTime>
<EndingDateTime>2009-01-05T05:30:30-05:00</EndingDateTime>
</RangeDateTime>
</Temporal>
<Spatial>
<GranuleLocality>
<LocalityValue>LocalityValue-0</LocalityValue>
<LocalityValue>LocalityValue-1</LocalityValue>
```

```
<LocalityValue>LocalityValue-2</LocalityValue>
<LocalityValue>LocalityValue-3</LocalityValue>
</GranuleLocality>
<VerticalSpatialDomains>
<VerticalSpatialDomain>
<Type>Type-4</Type>
<Value>Value-5</Value>
</VerticalSpatialDomain>
<VerticalSpatialDomain>
<Type>Type-6</Type>
<Value>Value-7</Value>
</VerticalSpatialDomain>
<VerticalSpatialDomain>
<Type>Type-8</Type>
<Value>Value-9</Value>
</VerticalSpatialDomain>
<VerticalSpatialDomain>
<Type>Type-10</Type>
<Value>Value-11</Value>
</VerticalSpatialDomain>
</VerticalSpatialDomains>
<HorizontalSpatialDomain>
<ZonelIdentifier>ZonelIdentifier</ZonelIdentifier>
<Geometry>
<Point>
<PointLongitude>-21.4</PointLongitude>
<PointLatitude>-15.4</PointLatitude>
</Point>
<Point>
<PointLongitude>-21.4</PointLongitude>
<PointLatitude>-16.4</PointLatitude>
</Point>
<Point>
<PointLongitude>-21.4</PointLongitude>
<PointLatitude>-17.4</PointLatitude>
</Point>
<BoundingRectangle>
```

```
<WestBoundingCoordinate>-163</WestBoundingCoordinate>
<NorthBoundingCoordinate>41</NorthBoundingCoordinate>
<EastBoundingCoordinate>-156</EastBoundingCoordinate>
<SouthBoundingCoordinate>40</SouthBoundingCoordinate>
</BoundingRectangle>
<BoundingRectangle>
<WestBoundingCoordinate>-163</WestBoundingCoordinate>
<NorthBoundingCoordinate>39</NorthBoundingCoordinate>
<EastBoundingCoordinate>-156</EastBoundingCoordinate>
<SouthBoundingCoordinate>38</SouthBoundingCoordinate>
</BoundingRectangle>
<BoundingRectangle>
<WestBoundingCoordinate>-163</WestBoundingCoordinate>
<NorthBoundingCoordinate>37</NorthBoundingCoordinate>
<EastBoundingCoordinate>-156</EastBoundingCoordinate>
<SouthBoundingCoordinate>36</SouthBoundingCoordinate>
</BoundingRectangle>
<GPolygon>
<Boundary>
<Point>
<PointLongitude>-79.4</PointLongitude>
<PointLatitude>39.6</PointLatitude>
</Point>
<Point>
<PointLongitude>-75.7</PointLongitude>
<PointLatitude>39.6</PointLatitude>
</Point>
<Point>
<PointLongitude>-75.2</PointLongitude>
<PointLatitude>38</PointLatitude>
</Point>
<Point>
<PointLongitude>-77.2</PointLongitude>
<PointLatitude>38.3</PointLatitude>
</Point>
<Point>
<PointLongitude>-77.5</PointLongitude>
```

```
<PointLatitude>39</PointLatitude>
</Point>
<Point>
<PointLongitude>-79.4</PointLongitude>
<PointLatitude>39.1</PointLatitude>
</Point>
</Boundary>
<ExclusiveZone>
<Boundary>
<Point>
<PointLongitude>-76.6</PointLongitude>
<PointLatitude>39.2</PointLatitude>
</Point>
<Point>
<PointLongitude>-76.1</PointLongitude>
<PointLatitude>39.2</PointLatitude>
</Point>
<Point>
<PointLongitude>-76.2</PointLongitude>
<PointLatitude>38.4</PointLatitude>
</Point>
<Point>
<PointLongitude>-76.5</PointLongitude>
<PointLatitude>38.4</PointLatitude>
</Point>
</Boundary>
</ExclusiveZone>
</GPolygon>
<Line>
<Point>
<PointLongitude>22.2</PointLongitude>
<PointLatitude>25.8</PointLatitude>
</Point>
<Point>
<PointLongitude>25</PointLongitude>
<PointLatitude>27.7</PointLatitude>
</Point>
```

```
<Point>
<PointLongitude>27.3</PointLongitude>
<PointLatitude>25.6</PointLatitude>
</Point>
</Line>
</Geometry>
</HorizontalSpatialDomain>
</Spatial>
<OrbitCalculatedSpatialDomains>
<OrbitCalculatedSpatialDomain>
<OrbitalModelName>OrbitalModelName</OrbitalModelName>
<OrbitNumber>0</OrbitNumber>
<StartOrbitNumber>0.0</StartOrbitNumber>
<StopOrbitNumber>0.0</StopOrbitNumber>
<EquatorCrossingLongitude>0.0</EquatorCrossingLongitude>
<EquatorCrossingDateTime>2009-01-05T05:30:30-05:00</EquatorCrossingDateTime>
</OrbitCalculatedSpatialDomain>
<OrbitCalculatedSpatialDomain>
<OrbitalModelName>OrbitalModelName</OrbitalModelName>
<OrbitNumber>0</OrbitNumber>
<StartOrbitNumber>0.0</StartOrbitNumber>
<StopOrbitNumber>0.0</StopOrbitNumber>
<EquatorCrossingLongitude>0.0</EquatorCrossingLongitude>
<EquatorCrossingDateTime>2009-01-05T05:30:30-05:00</EquatorCrossingDateTime>
</OrbitCalculatedSpatialDomain>
</OrbitCalculatedSpatialDomains>
<MeasuredParameters>
<MeasuredParameter>
<ParameterName>ParameterName-12</ParameterName>
<QAStats>
<QAPercentMissingData>3.1</QAPercentMissingData>
<QAPercentOutOfBoundsData>4.2</QAPercentOutOfBoundsData>
<QAPercentInterpolatedData>2.0</QAPercentInterpolatedData>
<QAPercentCloudCover>1.0</QAPercentCloudCover>
</QAStats>
<QAFlags>
<AutomaticQualityFlag>AutomaticQualityFlag</AutomaticQualityFlag>
```

```
<AutomaticQualityFlagExplanation>AutomaticQualityFlagExplanation</AutomaticQualityFlagExplanation>
<OperationalQualityFlag>OperationalQualityFlag</OperationalQualityFlag>
OperationalQualityFlagExplanation>OperationalQualityFlagExplanation</OperationalQualityFlagExplanation>
<ScienceQualityFlag>ScienceQualityFlag</ScienceQualityFlag>
<ScienceQualityFlagExplanation>ScienceQualityFlagExplanation</ScienceQualityFlagExplanation>
</QAFlags>
</MeasuredParameter>
<MeasuredParameter>
<ParameterName>ParameterName-13</ParameterName>
<QAStats>
<QAPercentMissingData>3.1</QAPercentMissingData>
<QAPercentOutOfBoundsData>4.2</QAPercentOutOfBoundsData>
<QAPercentInterpolatedData>2.0</QAPercentInterpolatedData>
<QAPercentCloudCover>1.0</QAPercentCloudCover>
</QAStats>
<QAFlags>
<AutomaticQualityFlag>AutomaticQualityFlag</AutomaticQualityFlag>
<AutomaticQualityFlagExplanation>AutomaticQualityFlagExplanation</AutomaticQualityFlagExplanation>
<OperationalQualityFlag>OperationalQualityFlag</OperationalQualityFlag>
OperationalQualityFlagExplanation>OperationalQualityFlagExplanation</OperationalQualityFlagExplanation>
<ScienceQualityFlag>ScienceQualityFlag</ScienceQualityFlag>
<ScienceQualityFlagExplanation>ScienceQualityFlagExplanation</ScienceQualityFlagExplanation>
</QAFlags>
</MeasuredParameter>
<MeasuredParameters>
<Platforms>
<Platform>
<ShortName>Short Name-26</ShortName>
<Instruments>
<Instrument>
<ShortName>Short Name-29</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-30</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
```

```
<Name>Characteristic Name-31</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name-32</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-33</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-34</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name-35</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-36</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-37</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode-38</OperationMode>
<OperationMode>Operation Mode-39</OperationMode>
</OperationModes>
</Instrument>
```

```
<Instrument>
<ShortName>Short Name-52</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-53</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-54</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name-55</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-56</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-57</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name-58</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-59</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-60</Name>
<Value>Characteristic Value</Value>
</Characteristic>
```

```
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode-62</OperationMode>
<OperationMode>Operation Mode-61</OperationMode>
</OperationModes>
</Instrument>
</Instruments>
</Platform>
<Platform>
<ShortName>Short Name-75</ShortName>
<Instruments>
<Instrument>
<ShortName>Short Name-78</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-79</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-80</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name-81</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-82</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-83</Name>
<Value>Characteristic Value</Value>
</Characteristic>
```

```
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name-84</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-85</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-86</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode-88</OperationMode>
<OperationMode>Operation Mode-87</OperationMode>
</OperationModes>
</Instrument>
<Instrument>
<ShortName>Short Name-101</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-102</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-103</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
<Sensors>
<Sensor>
<ShortName>Short Name-104</ShortName>
<Characteristics>
```

```
<Characteristic>
<Name>Characteristic Name-105</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-106</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
<Sensor>
<ShortName>Short Name-107</ShortName>
<Characteristics>
<Characteristic>
<Name>Characteristic Name-108</Name>
<Value>Characteristic Value</Value>
</Characteristic>
<Characteristic>
<Name>Characteristic Name-109</Name>
<Value>Characteristic Value</Value>
</Characteristic>
</Characteristics>
</Sensor>
</Sensors>
<OperationModes>
<OperationMode>Operation Mode-110</OperationMode>
<OperationMode>Operation Mode-111</OperationMode>
</OperationModes>
</Instrument>
</Instruments>
</Platform>
</Platforms>
<Campaigns>
<Campaign>
<ShortName>Short Name-131</ShortName>
</Campaign>
</Campaign>
```

```
<ShortName>Short Name-132</ShortName>
</Campaign>
</Campaigns>
<AdditionalAttributes>
<AdditionalAttribute>
<Name>Attribute Name-124</Name>
<Values>
<Value>Value</Value>
</Values>
<DataType>DataType</DataType>
</AdditionalAttribute>
<AdditionalAttribute>
<Name>Attribute Name-125</Name>
<Values>
<Value>Value</Value>
</Values>
<DataType>DataType</DataType>
</AdditionalAttribute>
<AdditionalAttribute>
<Name>Attribute Name-126</Name>
<Values>
<Value>1.4</Value>
</Values>
<DataType>DataType</DataType>
</AdditionalAttribute>
</AdditionalAttributes>
<InputGranules>
<InputGranule>InputGranule-14</InputGranule>
<InputGranule>InputGranule-15</InputGranule>
<InputGranule>InputGranule-16</InputGranule>
</InputGranules>
<TwoDCoordinateSystem>
<StartCoordinate1>0.0</StartCoordinate1>
<EndCoordinate1>0.0</EndCoordinate1>
<StartCoordinate2>0.0</StartCoordinate2>
<EndCoordinate2>0.0</EndCoordinate2>
<TwoDCoordinateSystemName>TwoDCoordinateSystemName</TwoDCoordinateSystemName>
```

```
</TwoDCoordinateSystem>
<Price>0.0</Price>
<OnlineAccessURLs>
<OnlineAccessURL>
<URL>http://www.google.com-17</URL>
<URLDescription>URLDescription</URLDescription>
<MimeType>MimeType</MimeType>
</OnlineAccessURL>
<OnlineAccessURL>
<URL>http://www.google.com-18</URL>
<URLDescription>URLDescription</URLDescription>
<MimeType>MimeType</MimeType>
</OnlineAccessURL>
</OnlineAccessURLs>
<OnlineResources>
<OnlineResource>
<URL>http://www.cnn.com-19</URL>
<Description>Description</Description>
<Type>Type</Type>
<MimeType>MimeType</MimeType>
</OnlineResource>
<OnlineResource>
<URL>http://www.cnn.com-20</URL>
<Description>Description</Description>
<Type>Type</Type>
<MimeType>MimeType</MimeType>
</OnlineResource>
</OnlineResources>
<Orderable>true</Orderable>
<DataFormat>JPEG Images</DataFormat>
<Visible>true</Visible>
<CloudCover>1.4</CloudCover>
<AssociatedBrowselImages>
<ProviderBrowseld>ProviderBrowseld-0</ProviderBrowseld>
<ProviderBrowseld>ProviderBrowseld-5</ProviderBrowseld>
</AssociatedBrowselImages>
</Granule>
```

```
</Granules>  
</GranuleMetaDataFile>
```

### H.3 BROWSE

```
<BrowseMetaDataFile>
<BrowseImages>
<BrowseImage>
<ProviderBrowseld>ProviderBrowseld-0</ProviderBrowseld>
<InsertTime>2009-01-05T05:30:30.550-05:00</InsertTime>
<LastUpdate>2009-01-05T05:30:30.550-05:00</LastUpdate>
<DeleteTime>2009-01-05T05:30:30.550-05:00</DeleteTime>
<FileName>fileName-1.jpg</FileName>
<FileSize>5795</FileSize>
<Description>Description-4</Description>
</BrowseImage>
<BrowseImage>
<ProviderBrowseld>ProviderBrowseld-5</ProviderBrowseld>
<InsertTime>2009-01-05T05:30:30.550-05:00</InsertTime>
<LastUpdate>2009-01-05T05:30:30.550-05:00</LastUpdate>
<DeleteTime>2009-01-05T05:30:30.550-05:00</DeleteTime>
<FileName>fileName-6.jpg</FileName>
<FileSize>8797</FileSize>
<Description>Description-9</Description>
</BrowseImage>
</BrowseImages>
</BrowseMetaDataFile>
```

# INDEX

- Browse
  - delete example, 100
  - DTD, 46, 96
  - insert example, 98
  - update example, 98
- Catalog
  - GetCatalogOrderInformatinon, 6
  - introduction to the term, 3
  - items, 6
- Collection
  - basic collection information, 14
  - introduction to the term, 3
  - required elements, 14
- Collections
  - delete items, 49, 91
- Data access rules
  - conditions, 70
  - group, 70
  - managing access, 70
  - PUMP, 70
  - restriction flag, 70
- DTD
  - browse, 46, 96
  - data not included in the DTD, 56
  - for collection metadata, 91
  - location for all ECHO DTDs, 3
  - metadata updated DTD*, 49, 87
  - separate DTDs for updating granules, collections, 3
- ECHO
  - concept, 2
  - design, 2
  - entities, 5
  - features, 2
  - graphical representation of concept, 2
- Elements
  - required and additional elements for a collection, 14
- Error handling
- catching exceptions, 102
- types of errors, 102
- Error message
  - data that violates data constraints, 63
  - example of granule associated with nonexistent collection, 65
  - example of return from invalid deletion, 64
  - granule associated with nonexistent collection, 65
  - invalid deletion, 64
- Error messages, 56
  - example of Ingest Summay Report, 60
  - example of XML validation error, 55
  - invalid XML file error, 55
  - non-XML, 58
- Exception catching, 102
- FTP, 3
  - ingest*, 3
  - location for ingest, 12
- GCMD, 18
- Granule
  - introduction to the term, 3
- Granules
  - delete items, 49, 91
- GUID
  - defined, 5
- Ingest
  - FTP location, 12
  - metadata mapping, 12
  - overview, 12
  - types of ingest information, 61
- Metadata mapping
  - ingest, 12
- Order options
  - defined, 8
- Validating your metadata
  - viewing dataset information, 68